1. (18 points)

a. (13 points)

Using Project 1 Relations/Tables **(including VIEWs)** create them in the Oracle DBMS database. Create **Project 2.sql** that contains the Commented SQL commands.

In **Project 2.doc** run one command at the time and take a screenshot of the commented command and its output and insert them in this document.

TABLE CUSTOMER

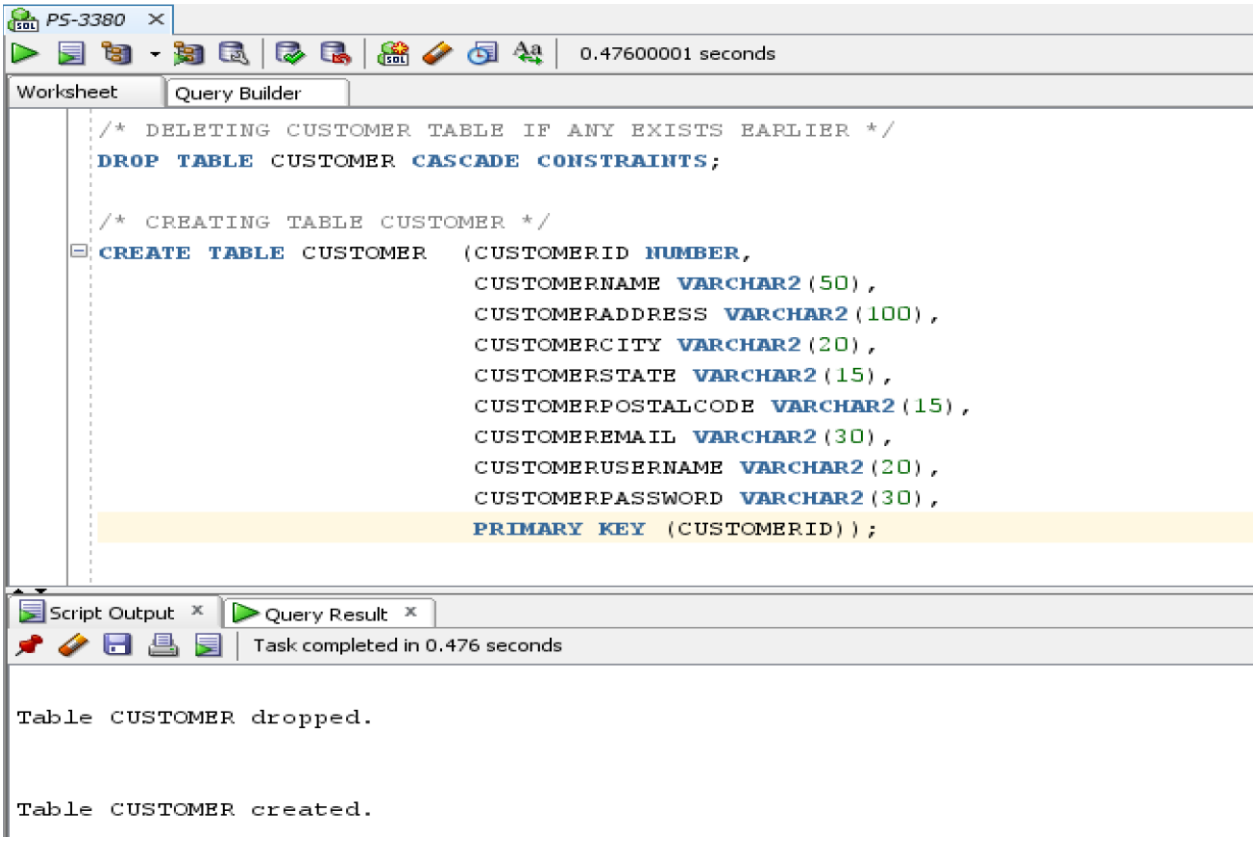


TABLE TERRITORY

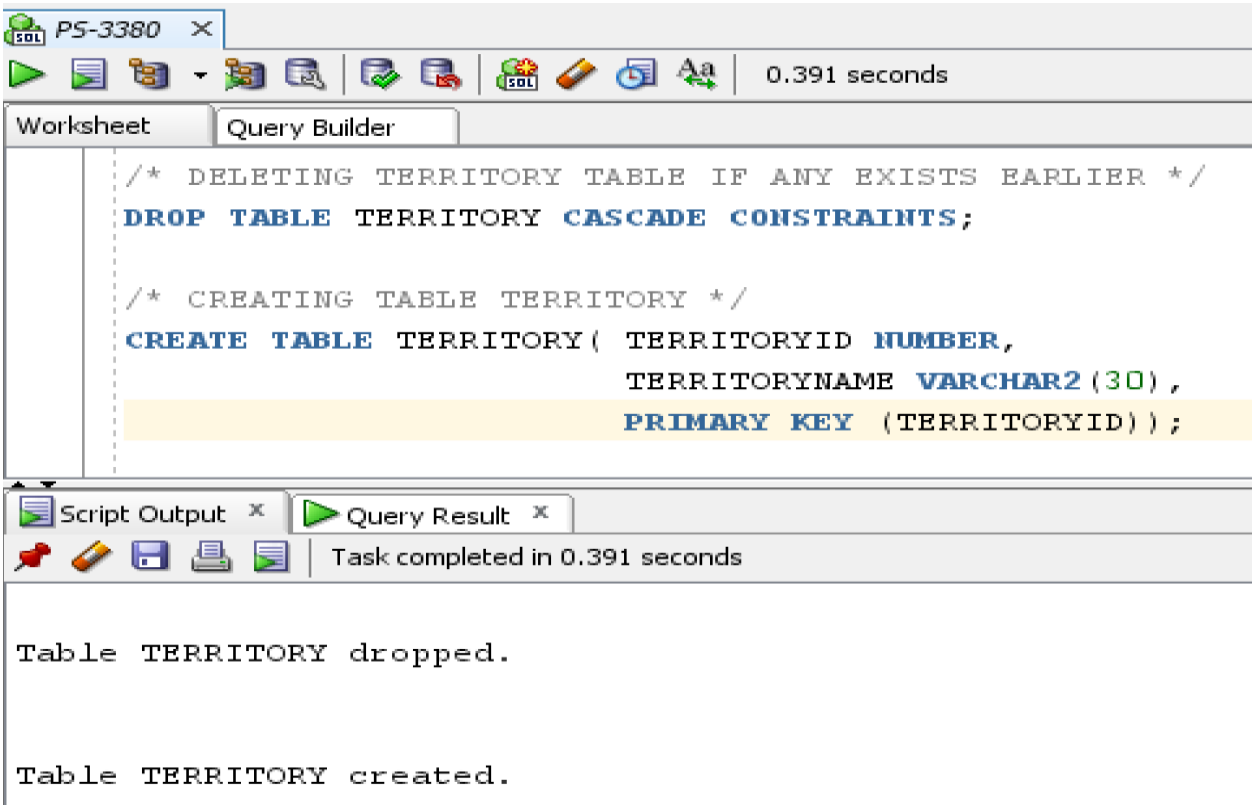


TABLE SALESPERSON

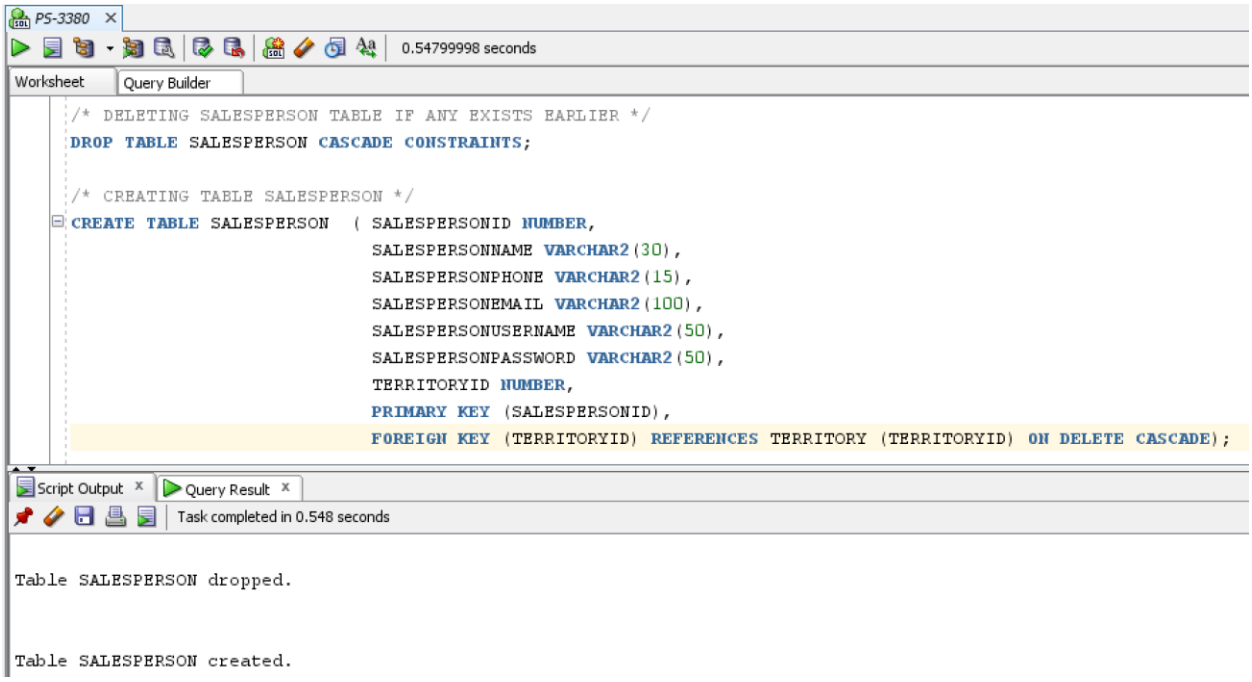


TABLE DOESBUSINESSIN

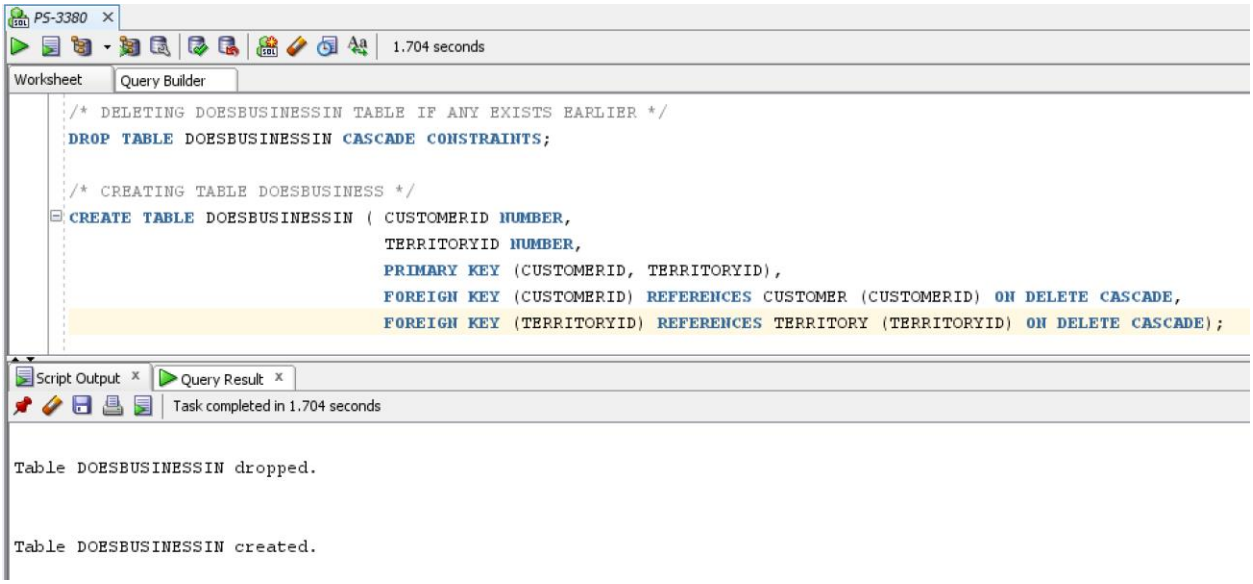


TABLE PRODUCTLINE

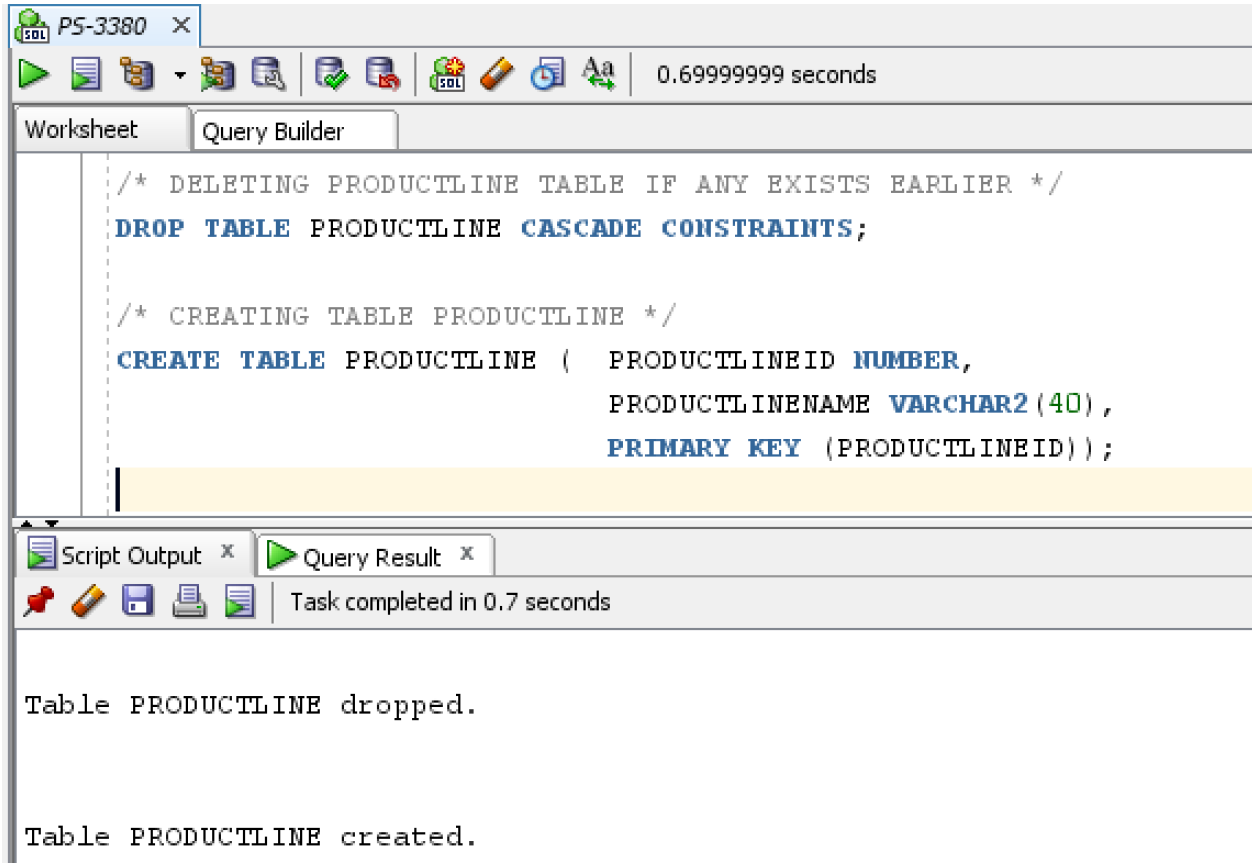


TABLE PRODUCT

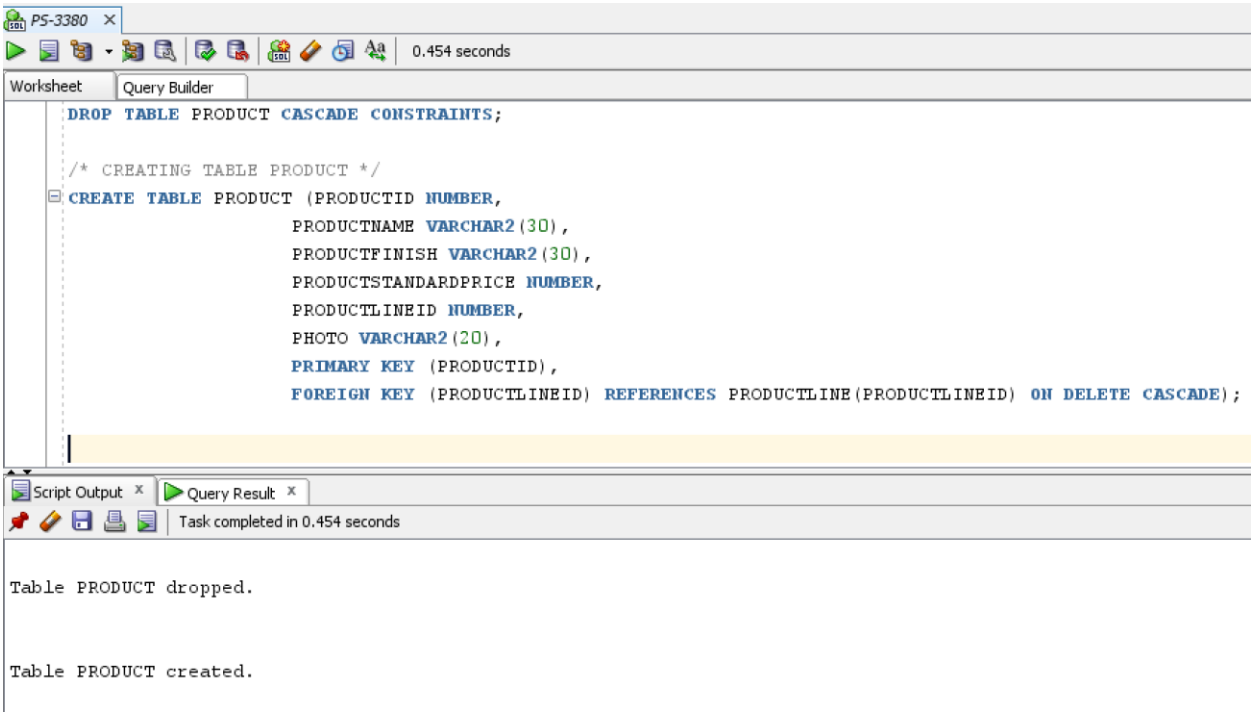


TABLE ORDERS

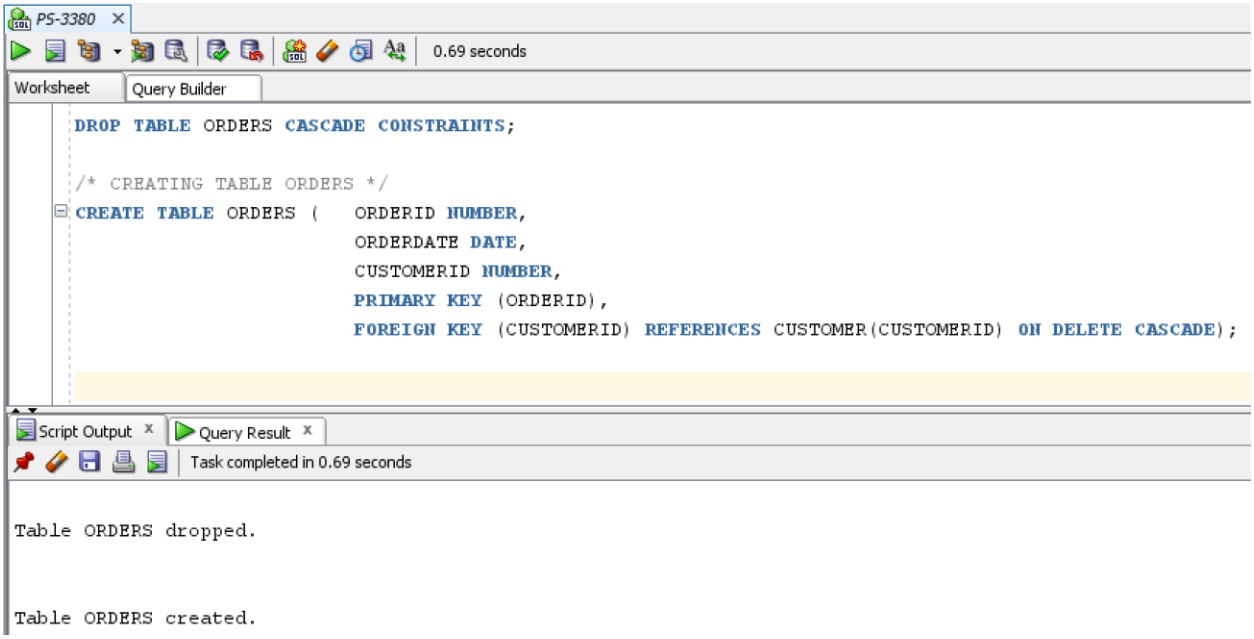


TABLE ORDERLINE

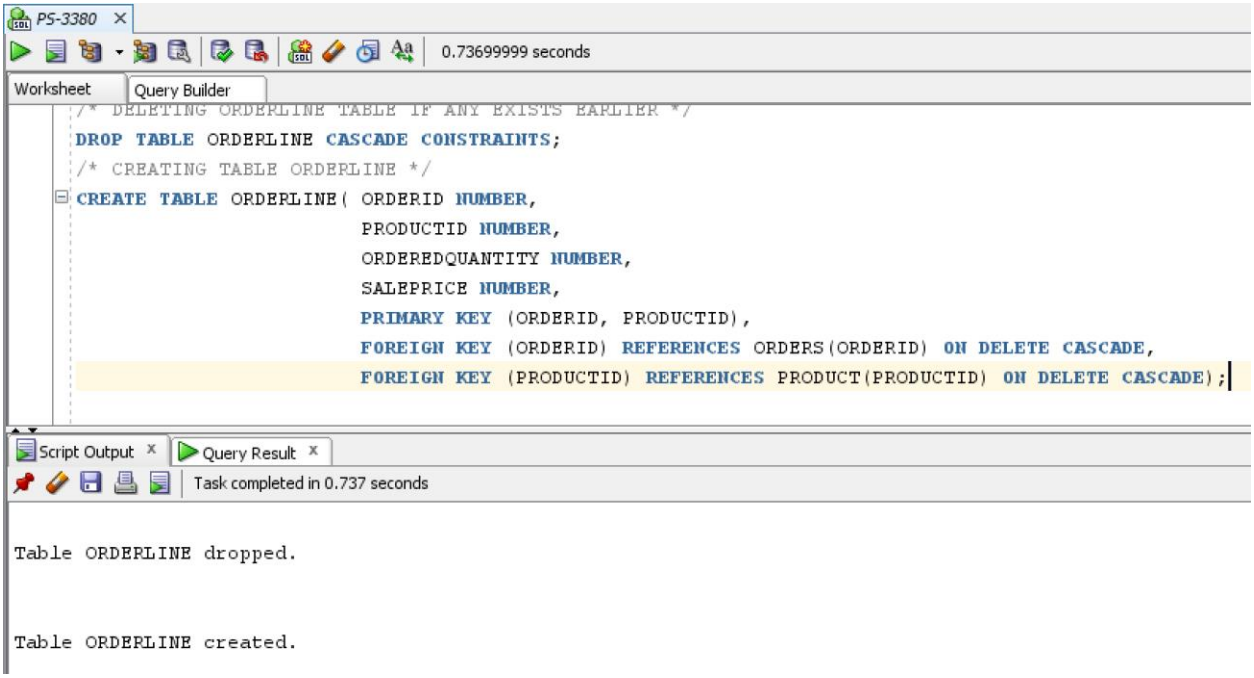
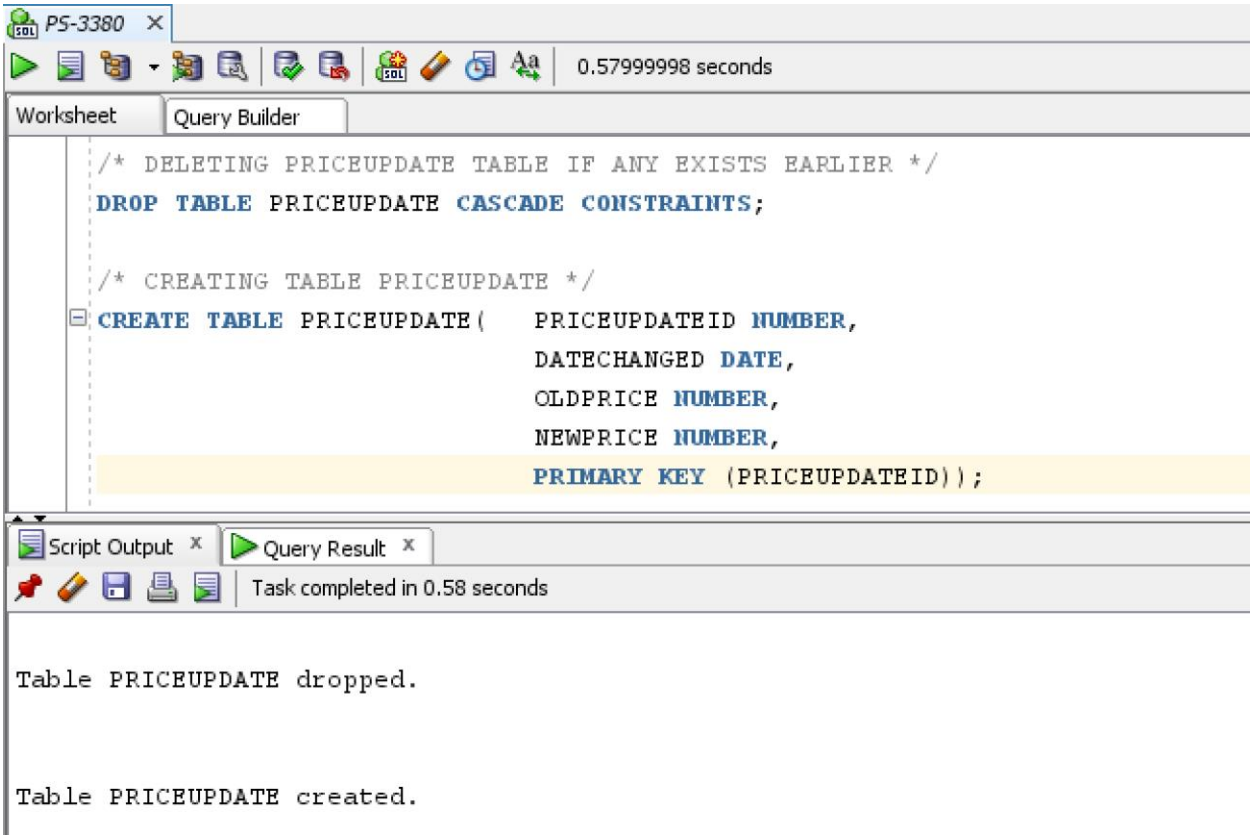


TABLE PRICEUPDATE



VIEW:

VIEW COMPARISON

PS-3380 | COMPARISION | 0.26100001 seconds

Worksheet | Query Builder

```
/* DELETE OR REPLACE VIEW COMPARISION IF ANY EXISTES EARLIER */
DROP VIEW COMPARISON CASCADE CONSTRAINTS;

/* CREATING OR REPLACING VIEW COMPARISON */
CREATE OR REPLACE VIEW COMPARISON(PRODUCTLINEID, PRODUCTID, PRODUCTNAME, SALESPRICE, ORDEREDQUANTITY, TOTALSALES) AS
SELECT PL.PRODUCTLINEID, P.PRODUCTID, P.PRODUCTNAME,OL.SALEPRICE, SUM(OL.ORDEREDQUANTITY),
(SUM(OL.ORDEREDQUANTITY)*OL.SALEPRICE)
FROM PRODUCTLINE PL, PRODUCT P, ORDERLINE OL
WHERE PL.PRODUCTLINEID = P.PRODUCTLINEID AND P.PRODUCTID = OL.PRODUCTID
GROUP BY PL.PRODUCTLINEID, P.PRODUCTID, P.PRODUCTNAME, OL.SALEPRICE
ORDER BY PL.PRODUCTLINEID;
```

Script Output | Query Result | Task completed in 0.261 seconds

```
View COMPARISON dropped.


View COMPARISON created.
```

VIEW TOTALVALUE

PS-3380 | COMPARISION | 0.41100001 seconds

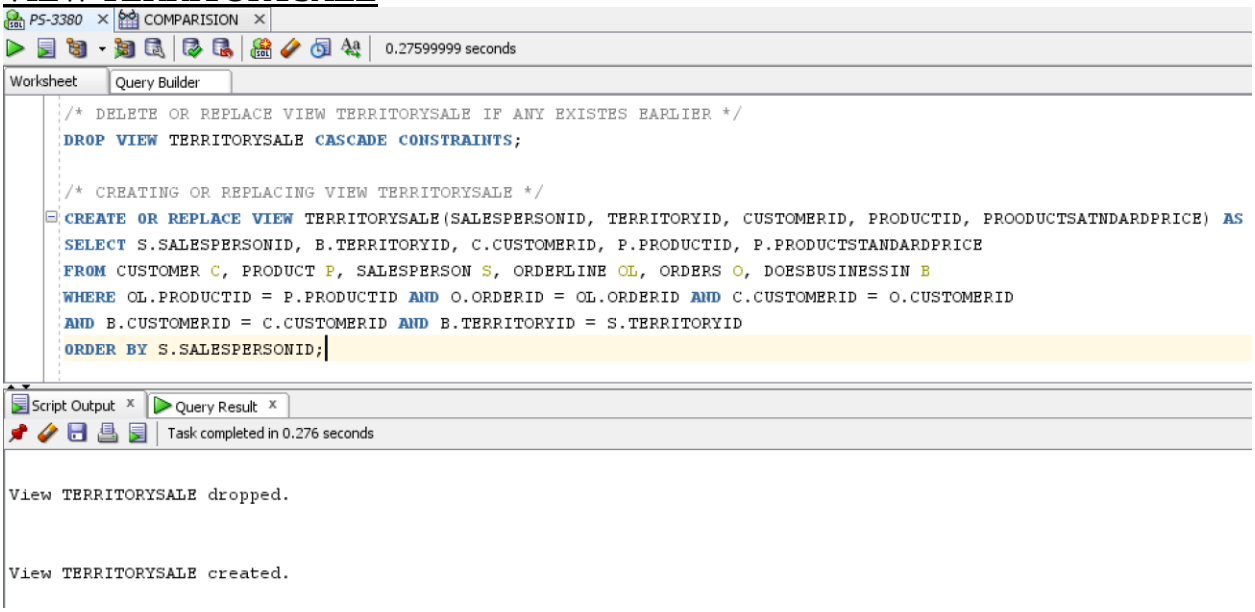
Worksheet | Query Builder

```
/* DELETE OR REPLACE VIEW TOTALVALUE IF ANY EXISTES EARLIER */
DROP VIEW TOTALVALUE CASCADE CONSTRAINTS;

/* CREATING OR REPLACING VIEW TOTALVALUE */
CREATE OR REPLACE VIEW TOTALVALUE(PRODUCTID, PRODUCTNAME, SALEPRICE, ORDEREDQUANTITY, TOTALSALES) AS
SELECT P.PRODUCTID, P.PRODUCTNAME, OL.SALEPRICE, SUM(OL.ORDEREDQUANTITY), (SUM(OL.ORDEREDQUANTITY)*OL.SALEPRICE)
FROM PRODUCT P, ORDERLINE OL
WHERE P.PRODUCTID = OL.PRODUCTID
GROUP BY P.PRODUCTID, P.PRODUCTNAME, OL.SALEPRICE
ORDER BY P.PRODUCTID;
```

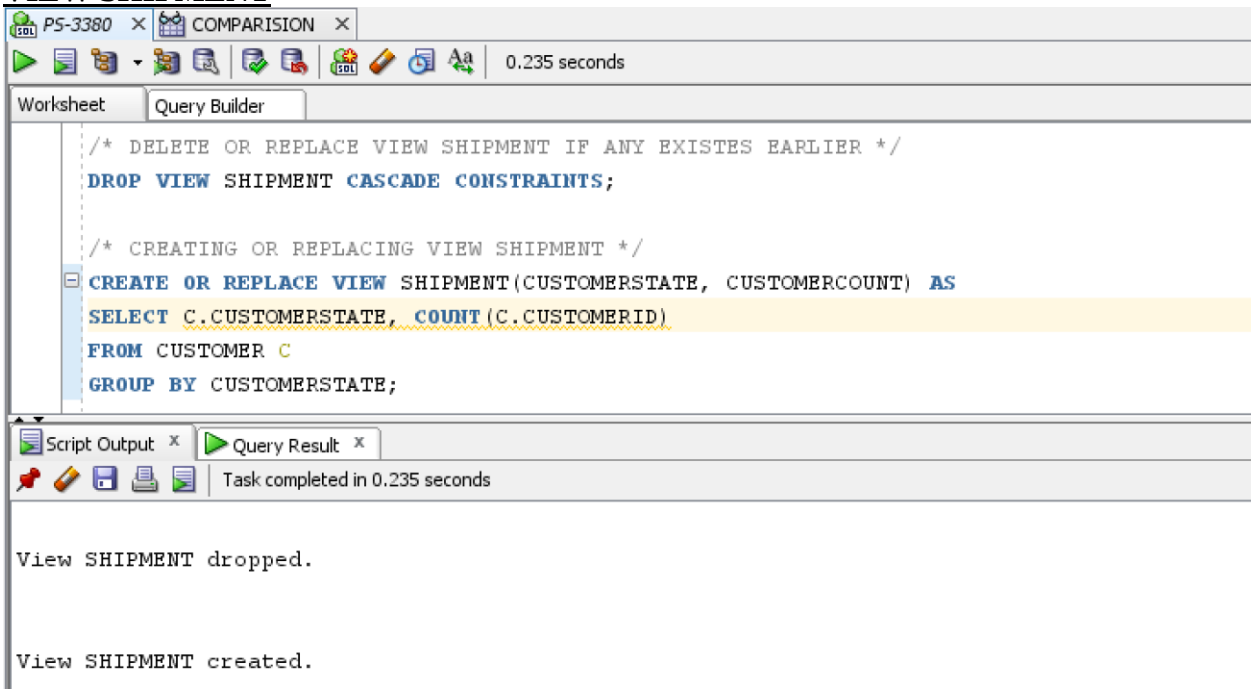
Script Output | Query Result | Task completed in 0.411 seconds

```
View TOTALVALUE dropped.


View TOTALVALUE created.
```

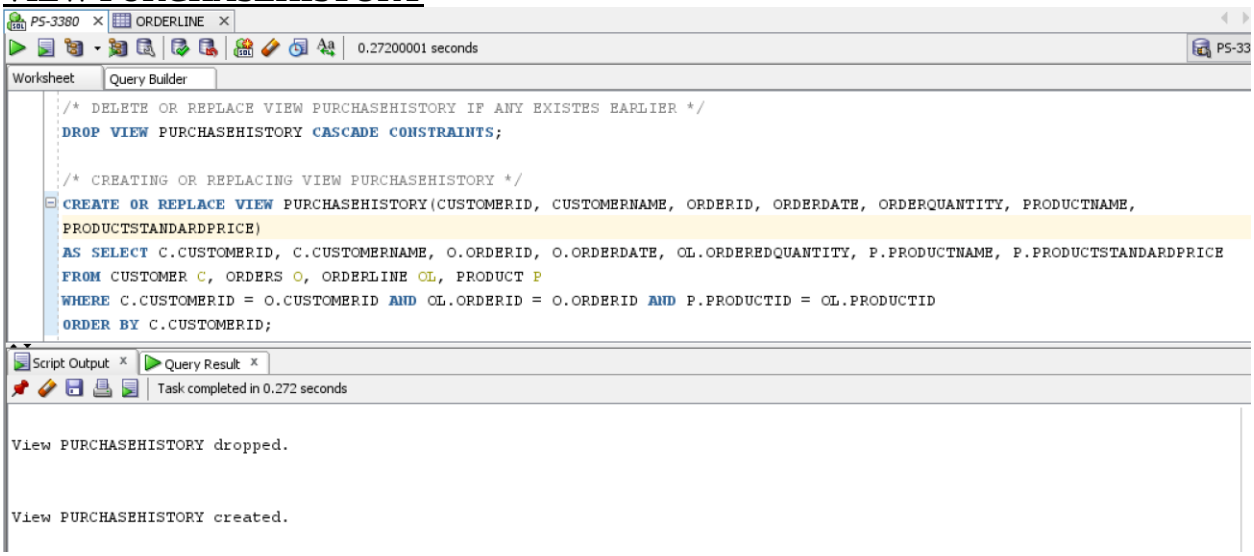
VIEW TERRITORYSALE



VIEW SHIPMENT



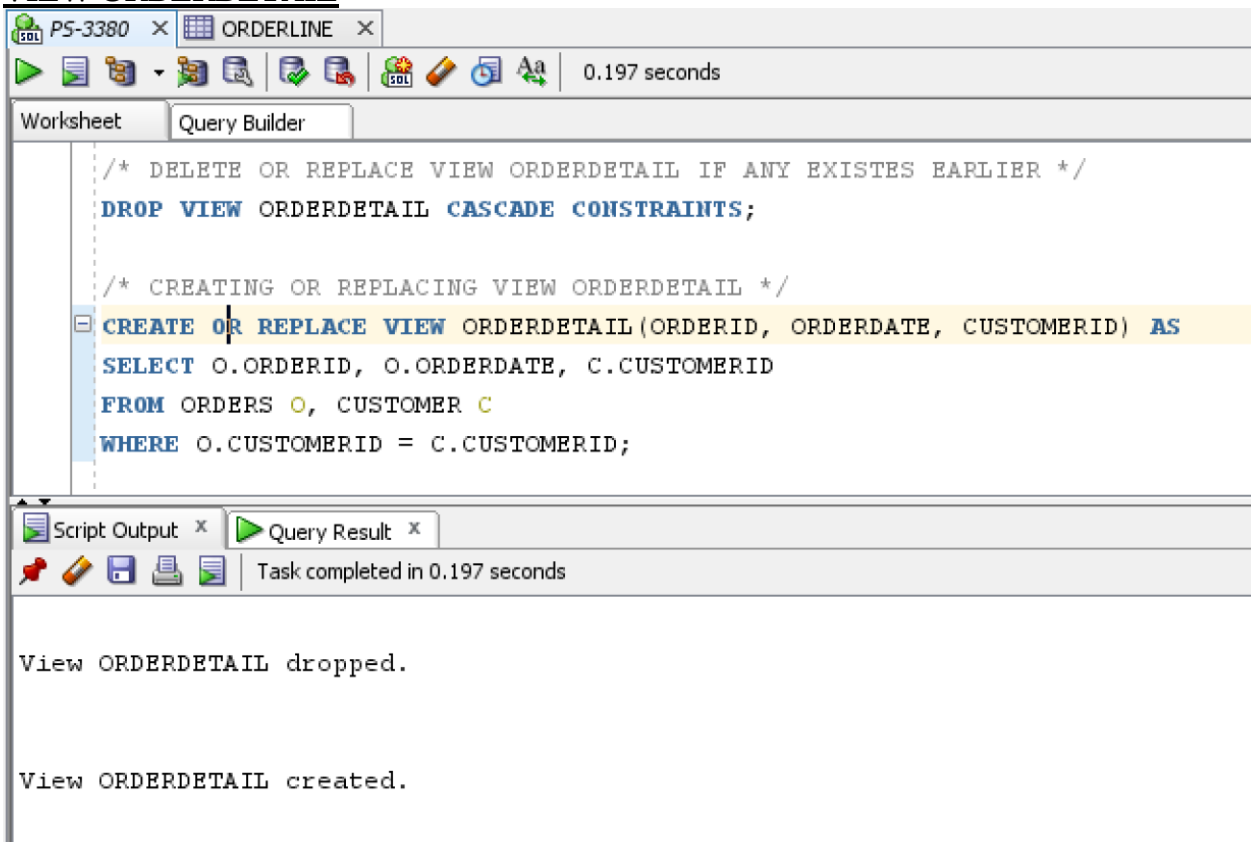
VIEW PURCHASEHISTORY



```
/* DELETE OR REPLACE VIEW PURCHASEHISTORY IF ANY EXISTES EARLIER */
DROP VIEW PURCHASEHISTORY CASCADE CONSTRAINTS;

/* CREATING OR REPLACING VIEW PURCHASEHISTORY */
CREATE OR REPLACE VIEW PURCHASEHISTORY(CUSTOMERID, CUSTOMERNAME, ORDERID, ORDERDATE, ORDERQUANTITY, PRODUCTNAME,
PRODUCTSTANDARDPRICE)
AS SELECT C.CUSTOMERID, C.CUSTOMERNAME, O.ORDERID, O.ORDERDATE, OL.ORDEREDQUANTITY, P.PRODUCTNAME, P.PRODUCTSTANDARDPRICE
FROM CUSTOMER C, ORDERS O, ORDERLINE OL, PRODUCT P
WHERE C.CUSTOMERID = O.CUSTOMERID AND OL.ORDERID = O.ORDERID AND P.PRODUCTID = OL.PRODUCTID
ORDER BY C.CUSTOMERID;
```

```
View PURCHASEHISTORY dropped.


View PURCHASEHISTORY created.
```

VIEW ORDERDETAIL



```
/* DELETE OR REPLACE VIEW ORDERDETAIL IF ANY EXISTES EARLIER */
DROP VIEW ORDERDETAIL CASCADE CONSTRAINTS;

/* CREATING OR REPLACING VIEW ORDERDETAIL */
CREATE OR REPLACE VIEW ORDERDETAIL(ORDERID, ORDERDATE, CUSTOMERID) AS
SELECT O.ORDERID, O.ORDERDATE, C.CUSTOMERID
FROM ORDERS O, CUSTOMER C
WHERE O.CUSTOMERID = C.CUSTOMERID;
```

```
View ORDERDETAIL dropped.


View ORDERDETAIL created.
```

TRIGGER

PS-3380 0.22400001 seconds

Worksheet | Query Builder

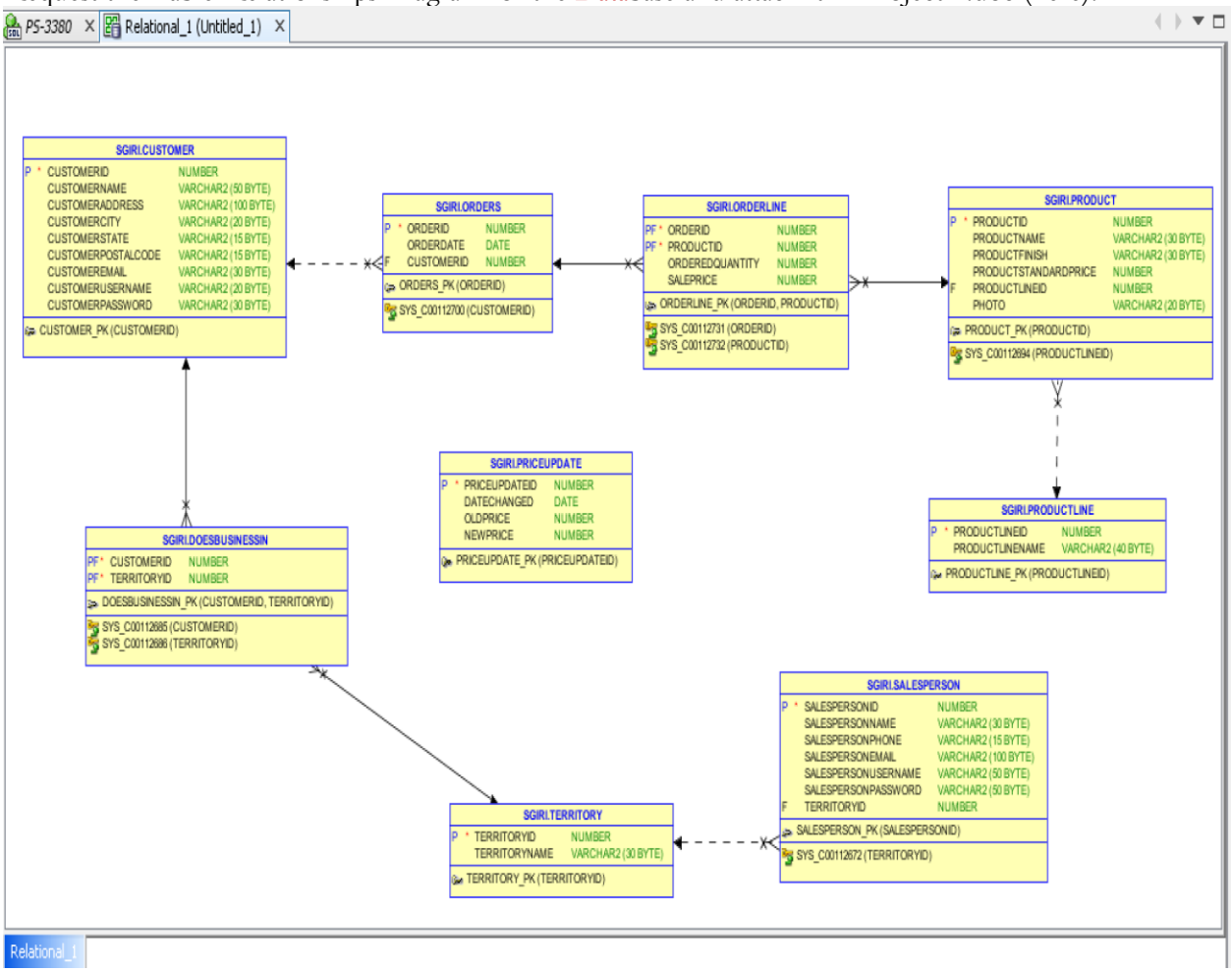
```
/* TRIGGER FOR STANDARDPRICEUPDATE */
CREATE OR REPLACE TRIGGER STANDARDPRICEUPDATE
AFTER UPDATE OF PRODUCTSTANDARDPRICE ON PRODUCT
FOR EACH ROW
DECLARE
        CURRENTDATE DATE;
BEGIN
    SELECT CURRENTDATE INTO CURRENTDATE
    FROM SYS.DUAL;

    INSERT INTO PRICEUPDATE (PRICEUPDATEID, DATECHANGED, OLDPRICE, NEWPRICE)
    VALUES (NULL,CURRENTDATE, :OLD.PRODUCTSTANDARDPRICE, :NEW.PRODUCTSTANDARDPRICE);
    DBMS_OUTPUT.PUT_LINE('SUCCESSFULLY ADDED');

END;
```

Script Output | Task completed in 0.224 seconds

```
Trigger STANDARDPRICEUPDATE compiled
```

b. (5 points)

Request the Table Relationships Diagram for the Database and attach it in Project 2.doc (here).



2. (40 points)

The Client has supplied the following data. Use **SQL** to **insert these data** in the Oracle DBMS database created in **Project 1**.

Customer

CustomerID, CustomerName, CustomerAddress, CustomerCity, CustomerState, CustomerPostalCode, CustomerEmail, CustomerUserName, CustomerPassword

1, 'Contemporary Casuals', '1355 S Hines Blvd', 'Gainesville', 'FL', '32601-2871'

2, 'Value Furnitures', '15145 S.W. 17th St.', 'Plano', 'TX', '75094-7734'

3, 'Home Furnishings', '1900 Allard Ave', 'Albany', 'NY', '12209-1125', **'homefurnishings?@gmail.com', 'CUSTOMER1', 'CUSTOMER1#'**

4, 'Eastern Furniture', '1925 Beltline Rd.', 'Carteret', 'NJ', '07008-3188'

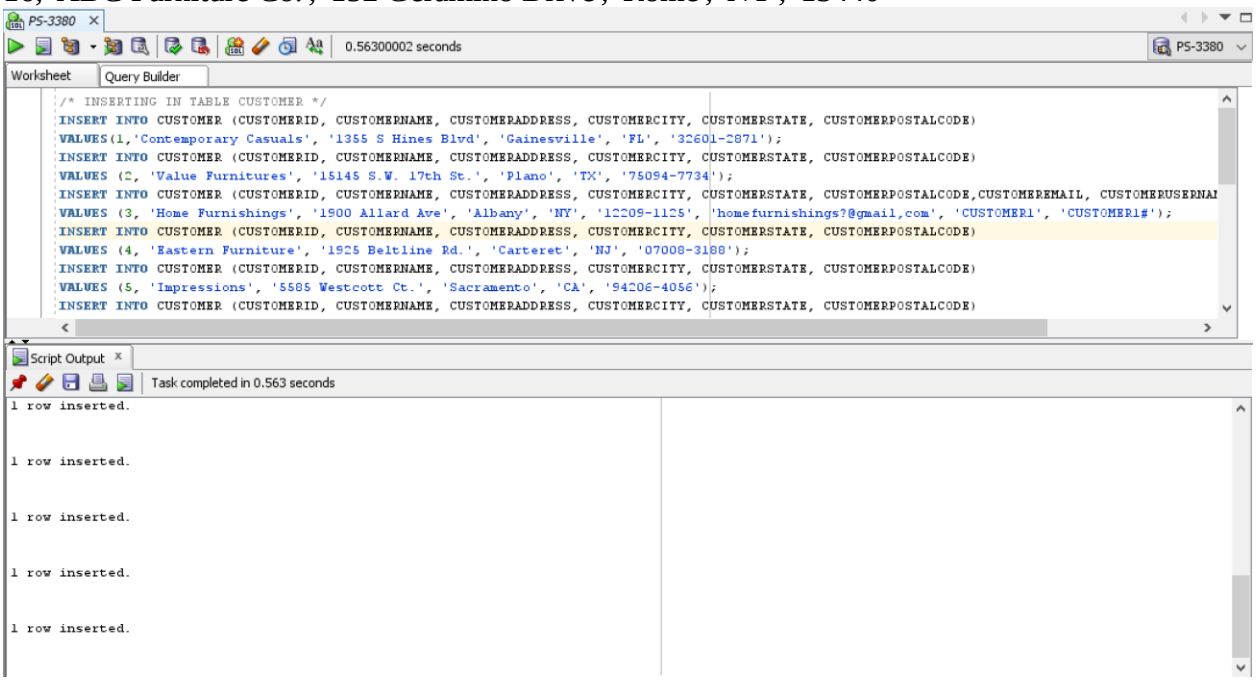
5, 'Impressions', '5585 Westcott Ct.', 'Sacramento', 'CA', '94206-4056'

6, 'Furniture Gallery', '325 Flatiron Dr.', 'Boulder', 'CO', '80514-4432'

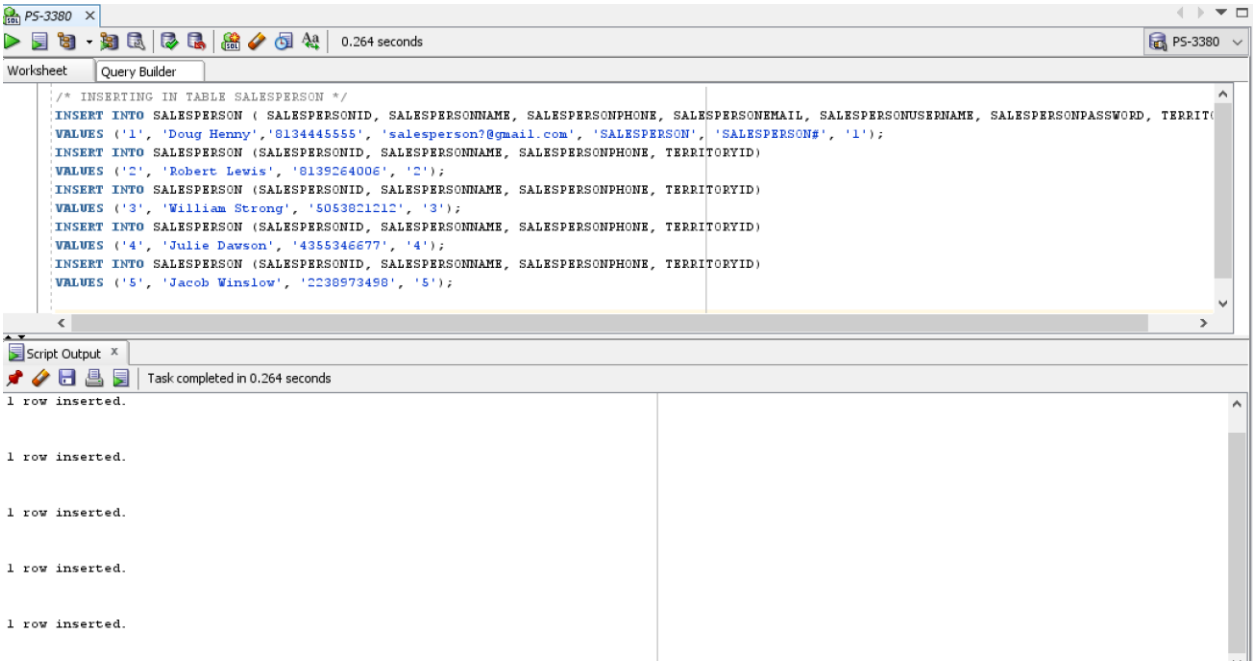
7, 'New Furniture', 'Palace Ave', 'Farmington', 'NM', ''

8, 'Dunkins Furniture', '7700 Main St', 'Syracuse', 'NY', '31590'

9, 'A Carpet', '434 Abe Dr', 'Rome', 'NY', '13440'
12, 'Flanigan Furniture', 'Snow Flake Rd', 'Ft Walton Beach', 'FL', '32548'
13, 'Ikards', '1011 S. Main St', 'Las Cruces', 'NM', '88001'
14, 'Wild Bills', 'Four Horse Rd', 'Oak Brook', 'Il', '60522'
15, 'Janet's Collection', 'Janet Lane', 'Virginia Beach', 'VA', '10012'
16, 'ABC Furniture Co.', '152 Geramino Drive', 'Rome', 'NY', '13440'



Salesperson
SalespersonID, SalespersonName, SalespersonPhone, SalespersonEmail, SalespersonUserName, SalespersonPassword,TerritoryID
1, 'Doug Henny', '8134445555', **'salesperson?@gmail.com'**, '**SALESPERSON**', '**SALESPERSON#**',1
2, 'Robert Lewis', '8139264006', '', '', '', 2
3, 'William Strong', '5053821212', '', '', '', 3
4, 'Julie Dawson', '4355346677', '', '', '', 4
5, 'Jacob Winslow', '2238973498', '', '', '', 5



Territory

TerritoryID, TerritoryName

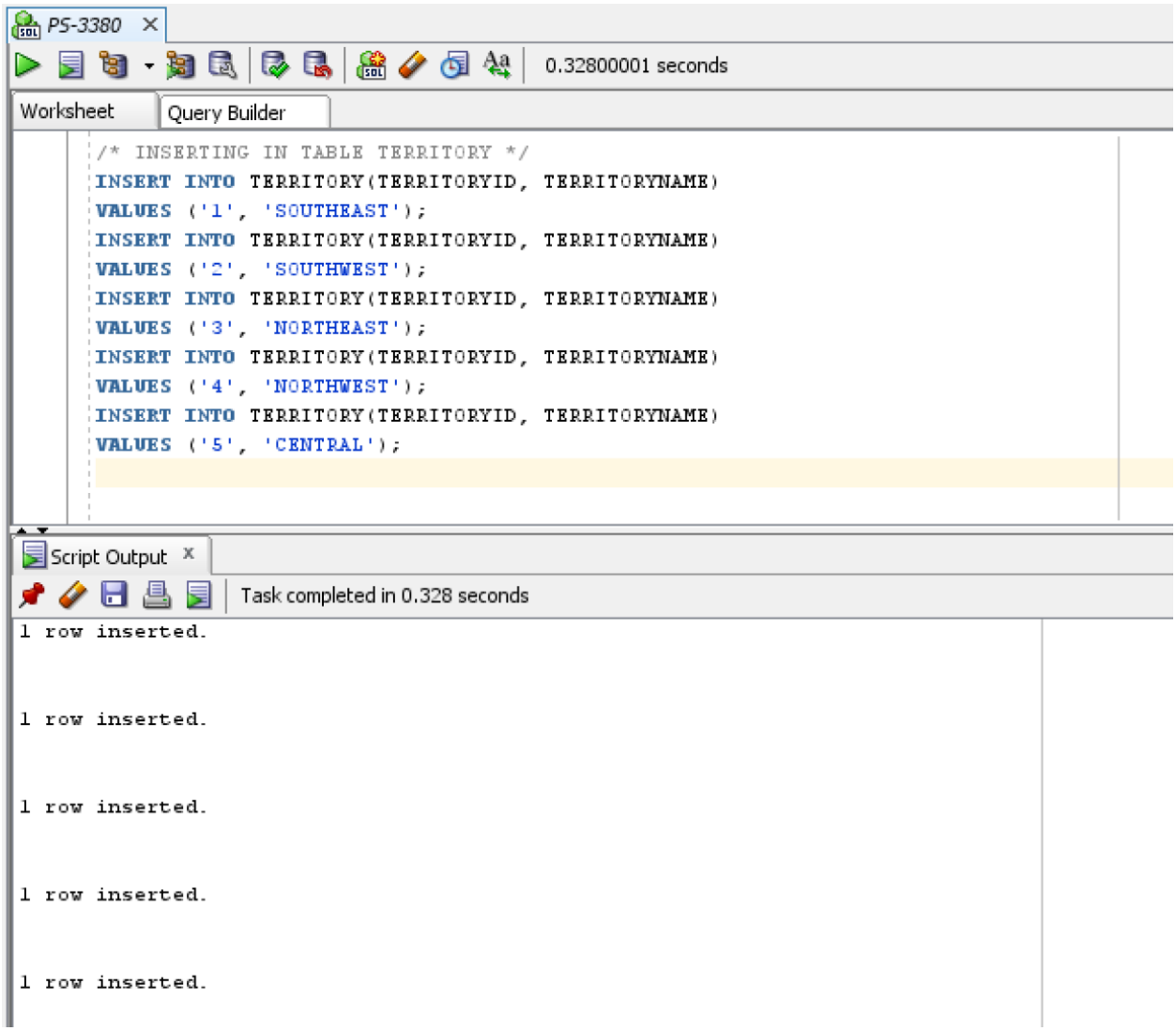
1, 'SouthEast'

2, 'SouthWest'

3, 'NorthEast'

4, 'NorthWest'

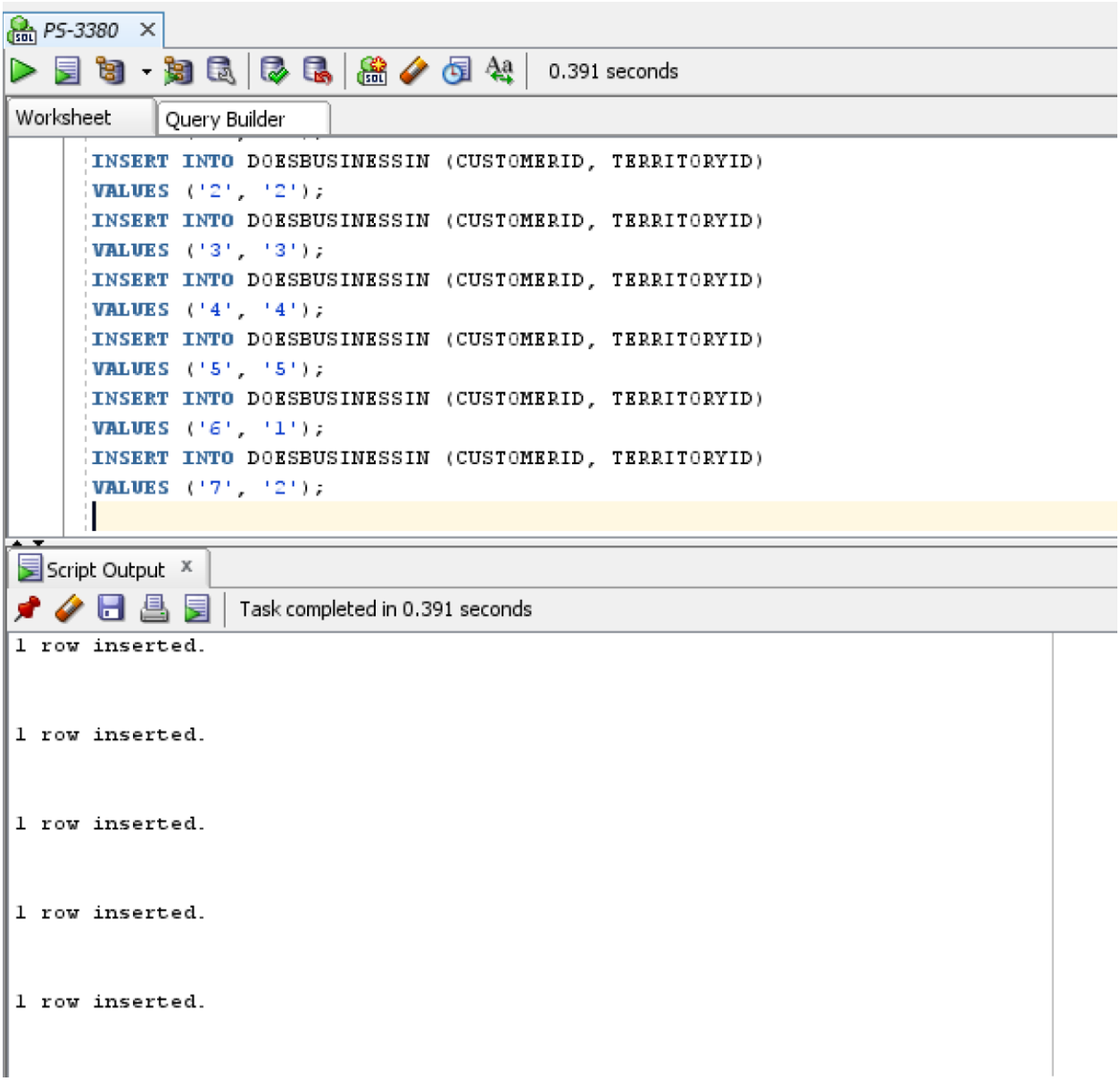
5, 'Central'



DoesBusinessIn

CustomerID, TerritoryID

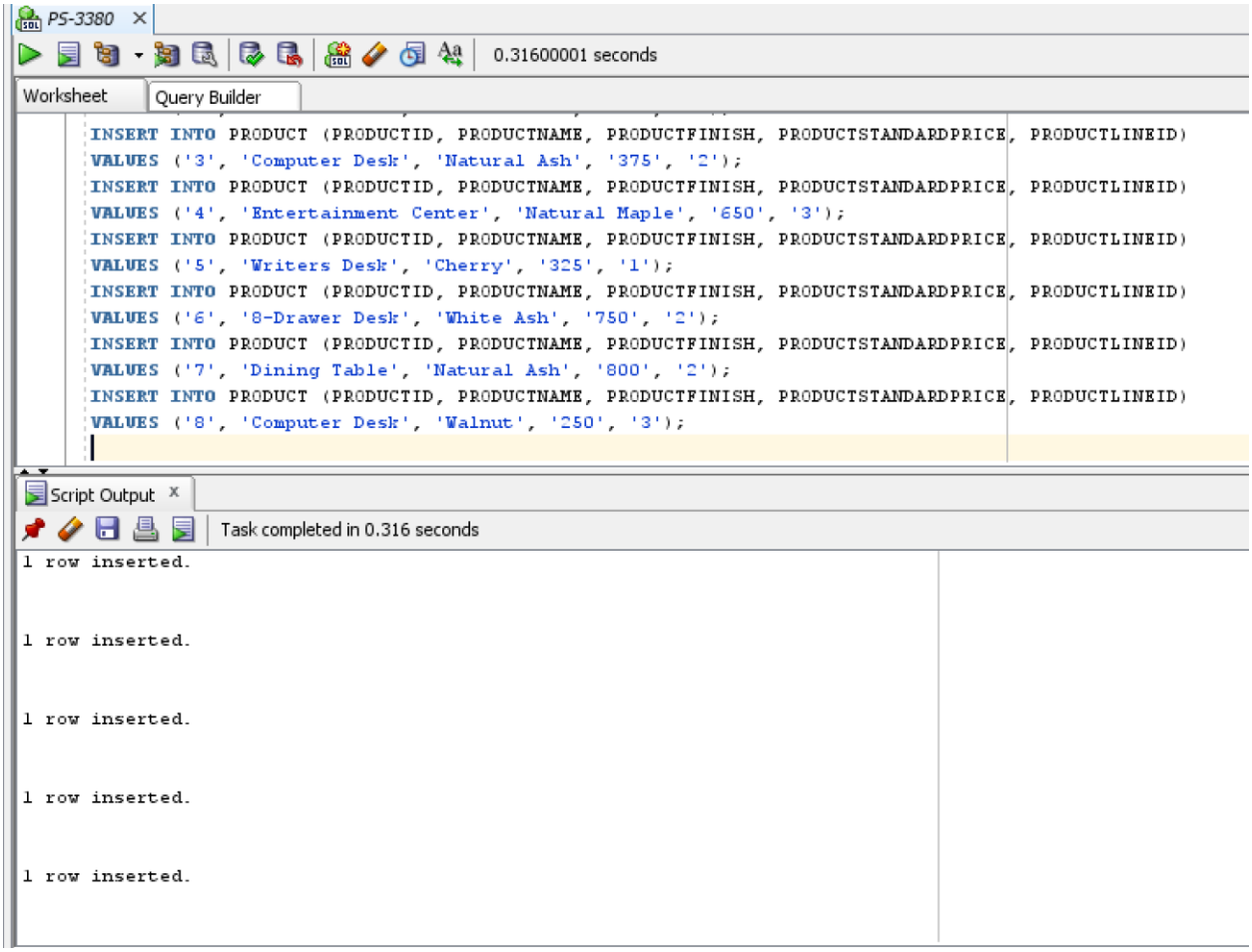
1, 1
2, 2
3, 3
4, 4
5, 5
6, 1
7, 2



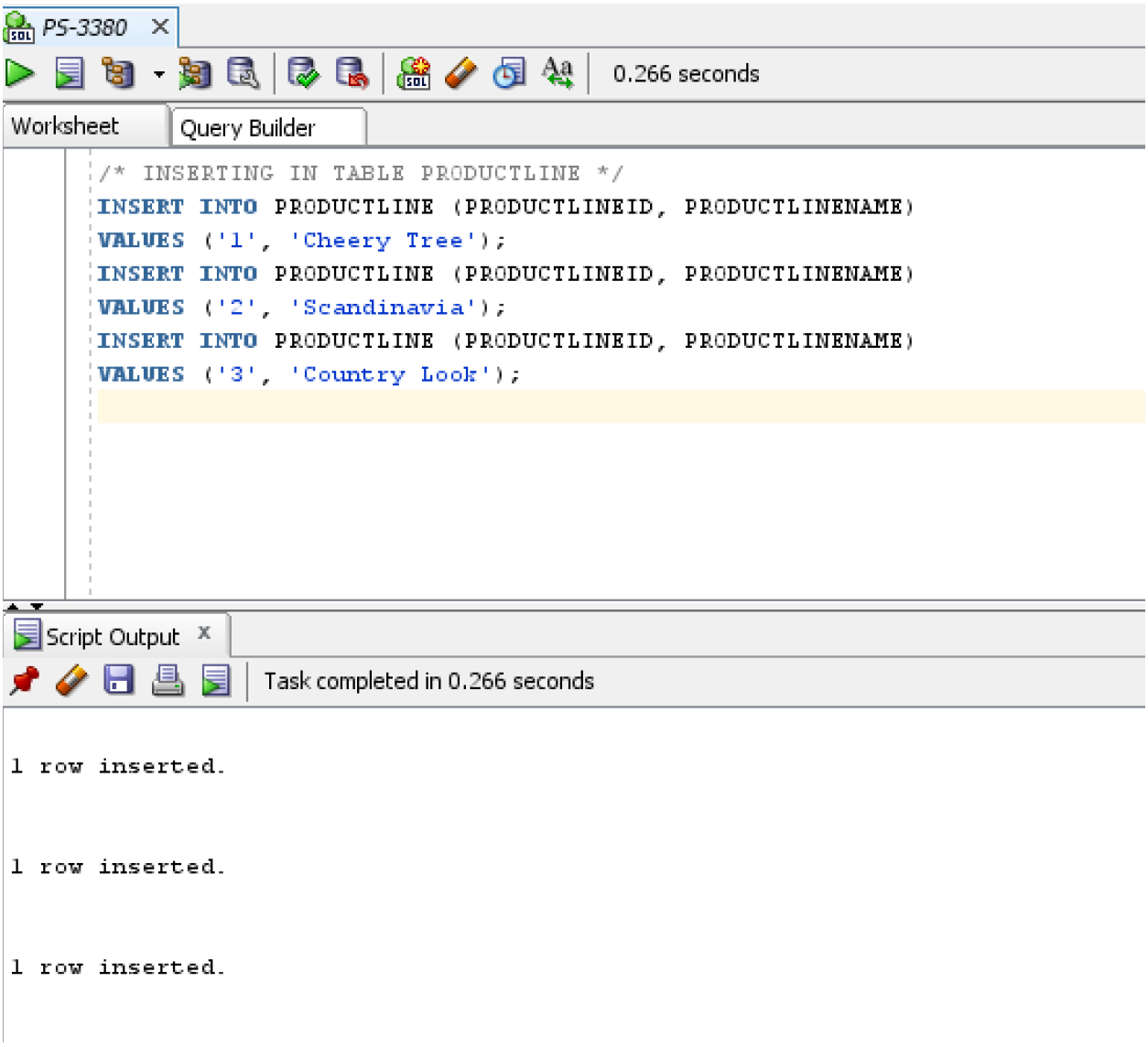
Product

ProductID, ProductName, ProductFinish, ProductStandardPrice, ProductLineID, **Photo**

1, 'End Table', 'Cherry', 175, 1, **table.jpg**
2, 'Coffee Table', 'Natural Ash', 200, 2
3, 'Computer Desk', 'Natural Ash', 375, 2
4, 'Entertainment Center', 'Natural Maple', 650, 3
5, 'Writers Desk', 'Cherry', 325, 1
6, '8-Drawer Desk', 'White Ash', 750, 2
7, 'Dining Table', 'Natural Ash', 800, 2
8, 'Computer Desk', 'Walnut', 250, 3



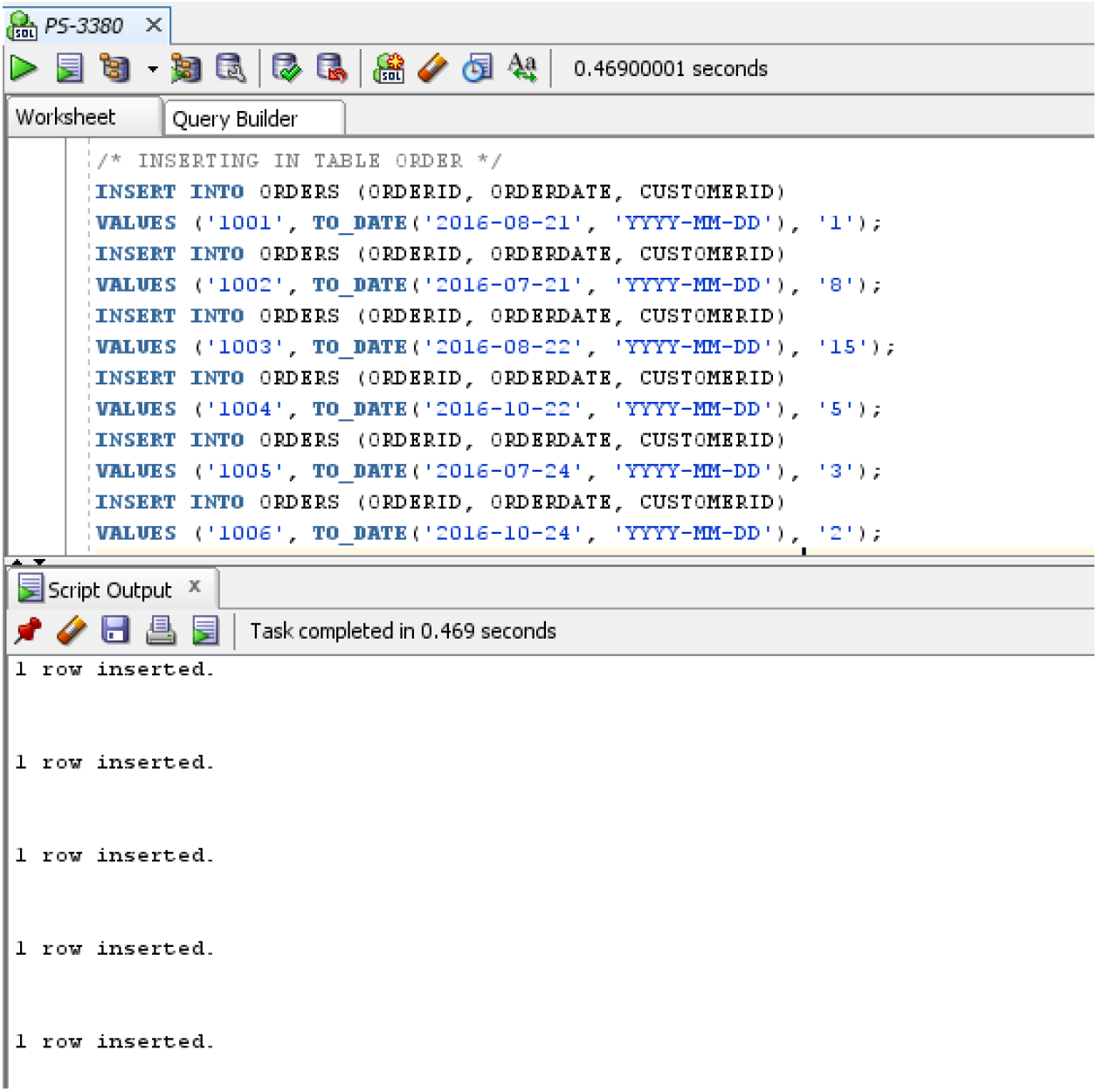
ProductLine
ProductLineID, ProductLineName
1, 'Cherry Tree'
2, 'Scandinavia'
3, 'Country Look'



Order

OrderID, OrderDate, CustomerID

1001, '21/Aug/16', 1
1002, '21/Jul/16', 8
1003, '22/ Aug/16', 15
1004, '22/Oct/16', 5
1005, '24/Jul/16', 3
1006, '24/Oct/16', 2
1007, '27/ Aug/16', 5
1008, **'30/Oct/16', 12**
1009, **'05/Nov/16', 4**
1010, **'05/Nov/16', 1**

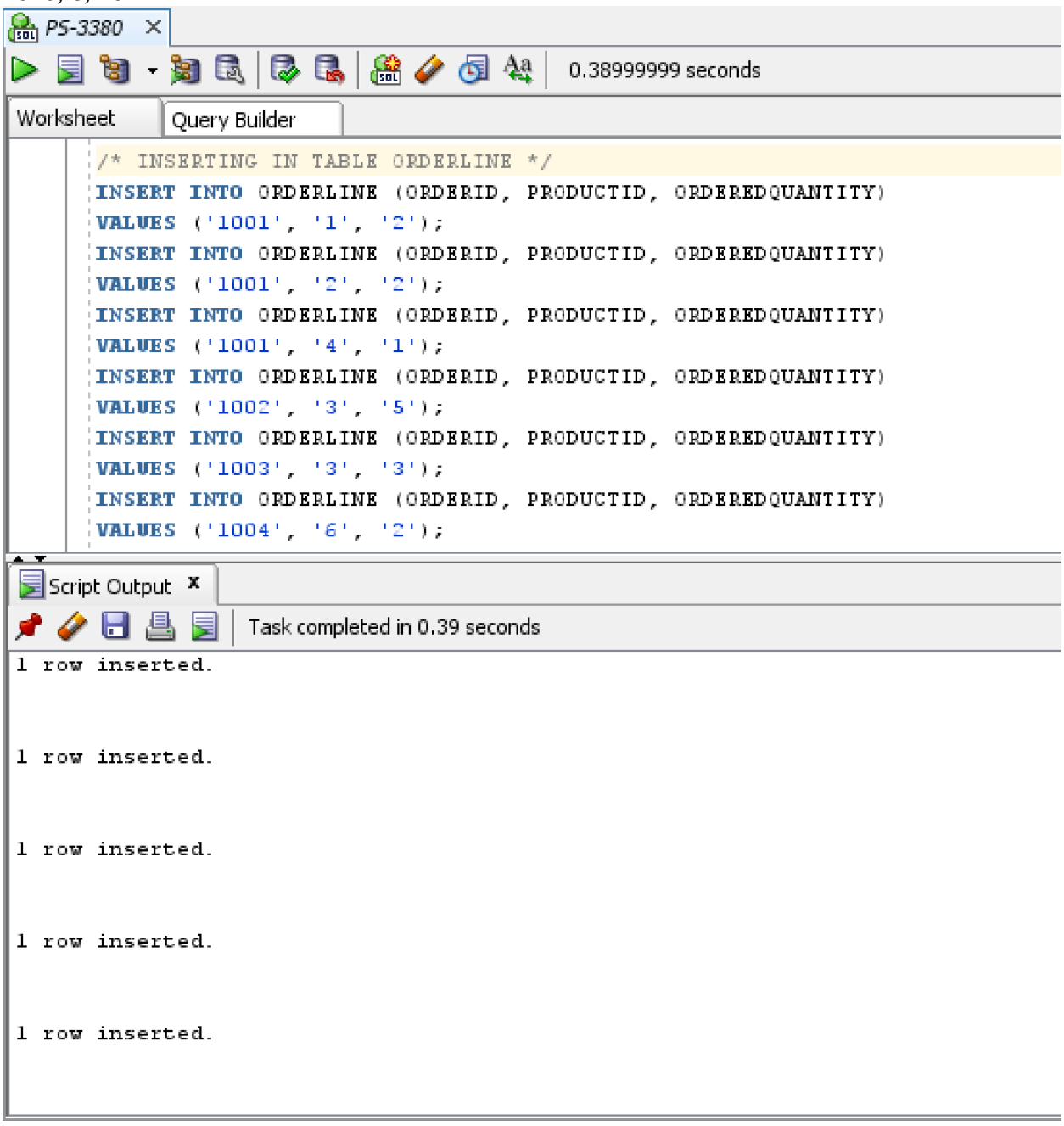


OrderLine

OrderID, ProductID, OrderedQuantity, SalePrice

1001, 1, 2
1001, 2, 2
1001, 4, 1
1002, 3, 5
1003, 3, 3
1004, 6, 2
1004, 8, 2
1005, 4, 4
1006, 4, 1
1006, 5, 2

1006, 7, 2
1007, 1, 3
1007, 2, 2
1008, 3, 3
1008, 8, 3
1009, 4, 2
1009, 7, 3
1010, 8, 10



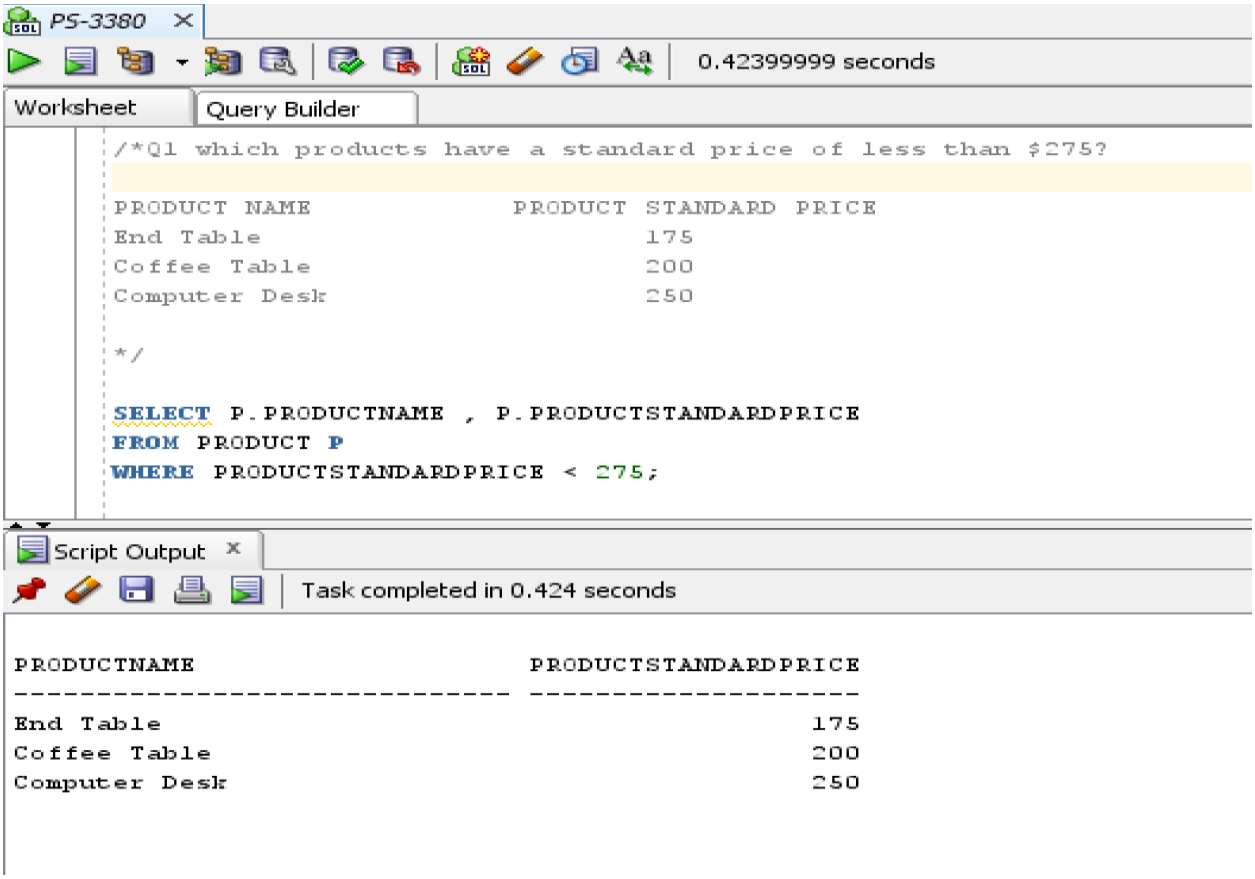
PriceUpdate
PriceUpdateID, DateChanged, OldPrice, NewPrice

3. (42 points)

The Client wishes that you create the **SQL** code for the following queries (please make sure you SQL COMMENT each query and produce the **screenshots for SQL COMMENT, SQL COMMAND, and SQL OUTPUT each, to be placed in Project 2.doc**):

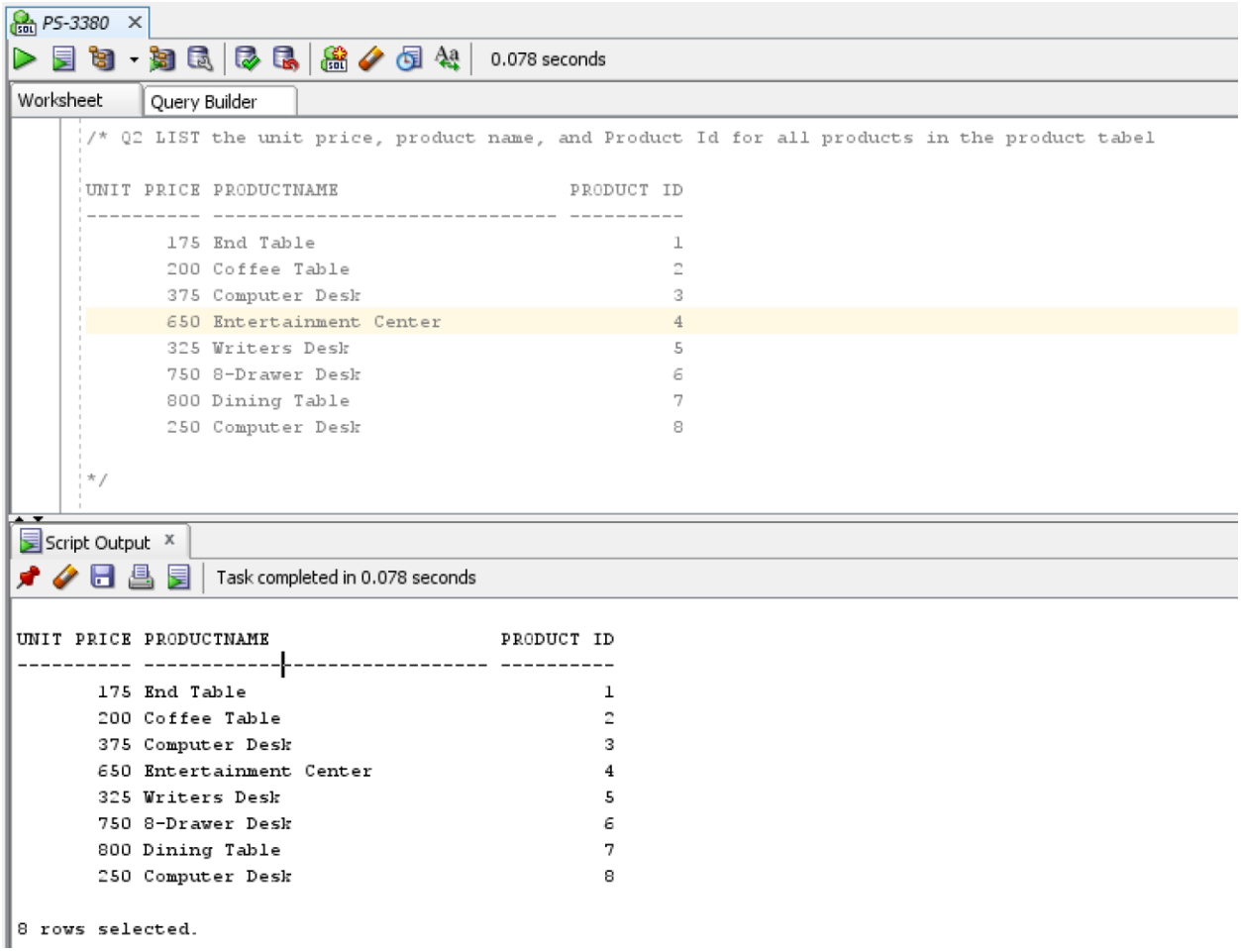
1. (3 points)

Which products have a standard price of less than $ 275?



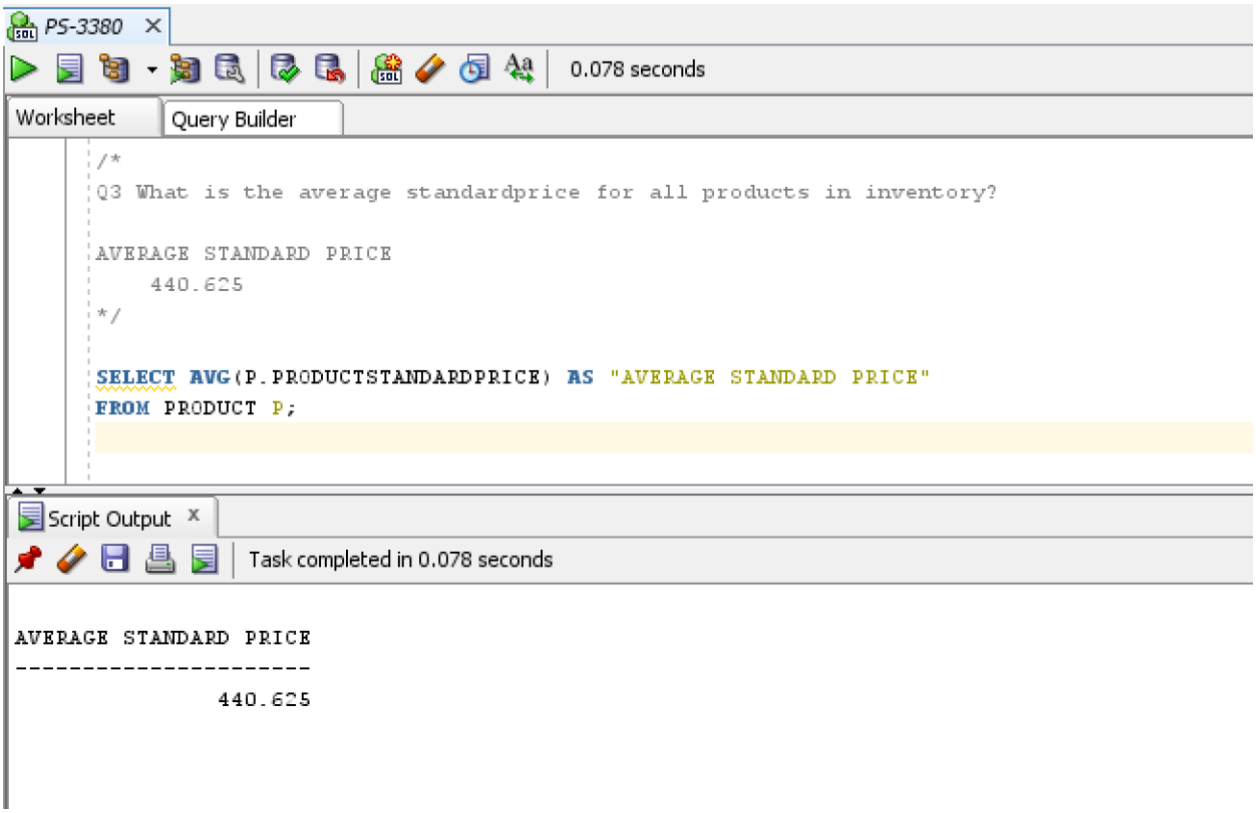
2. (3 points)

List the unit price, product name, and product ID for all products in the Product table.



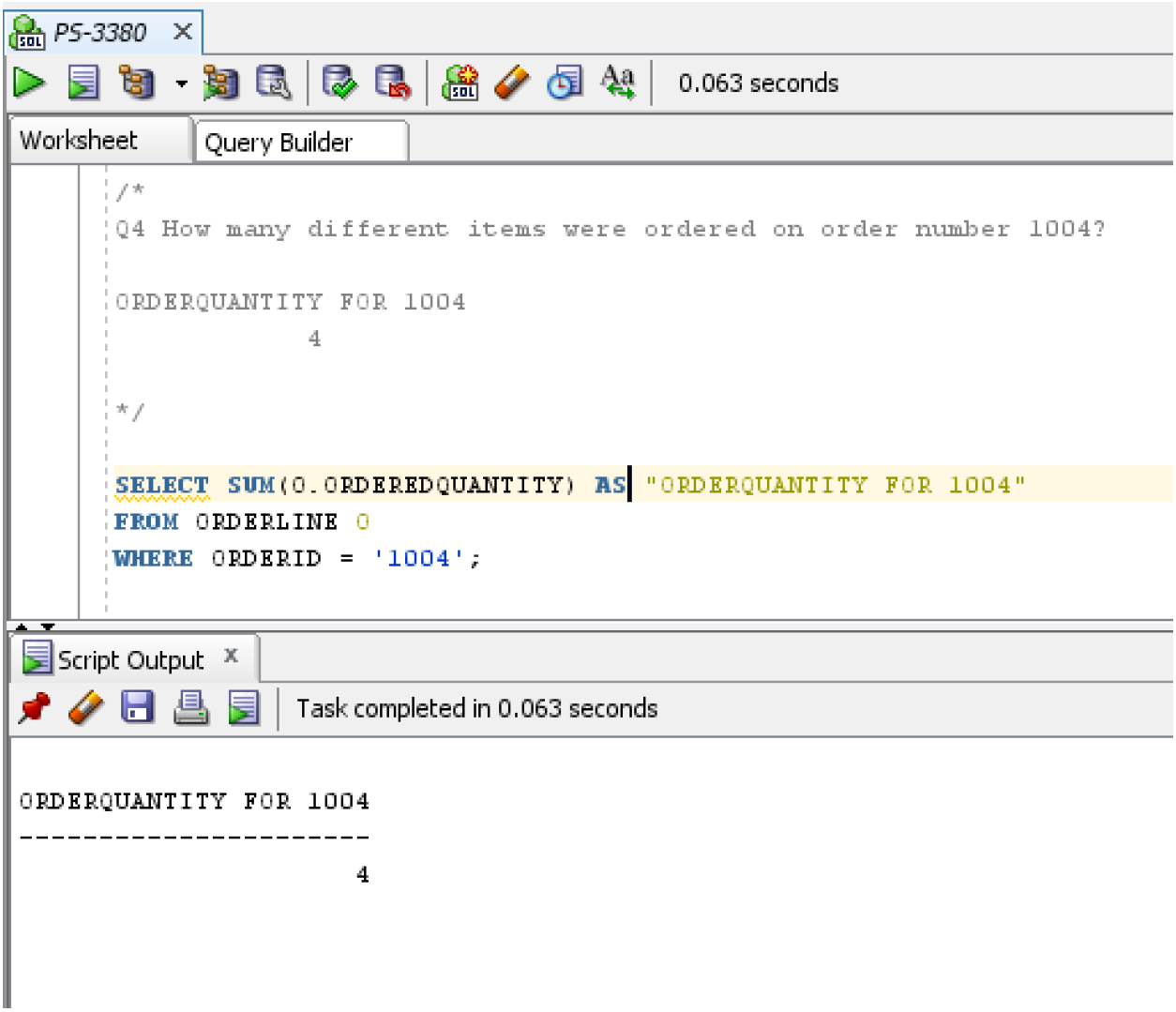
3. (3 points)

What is the average standard price for all products in inventory?



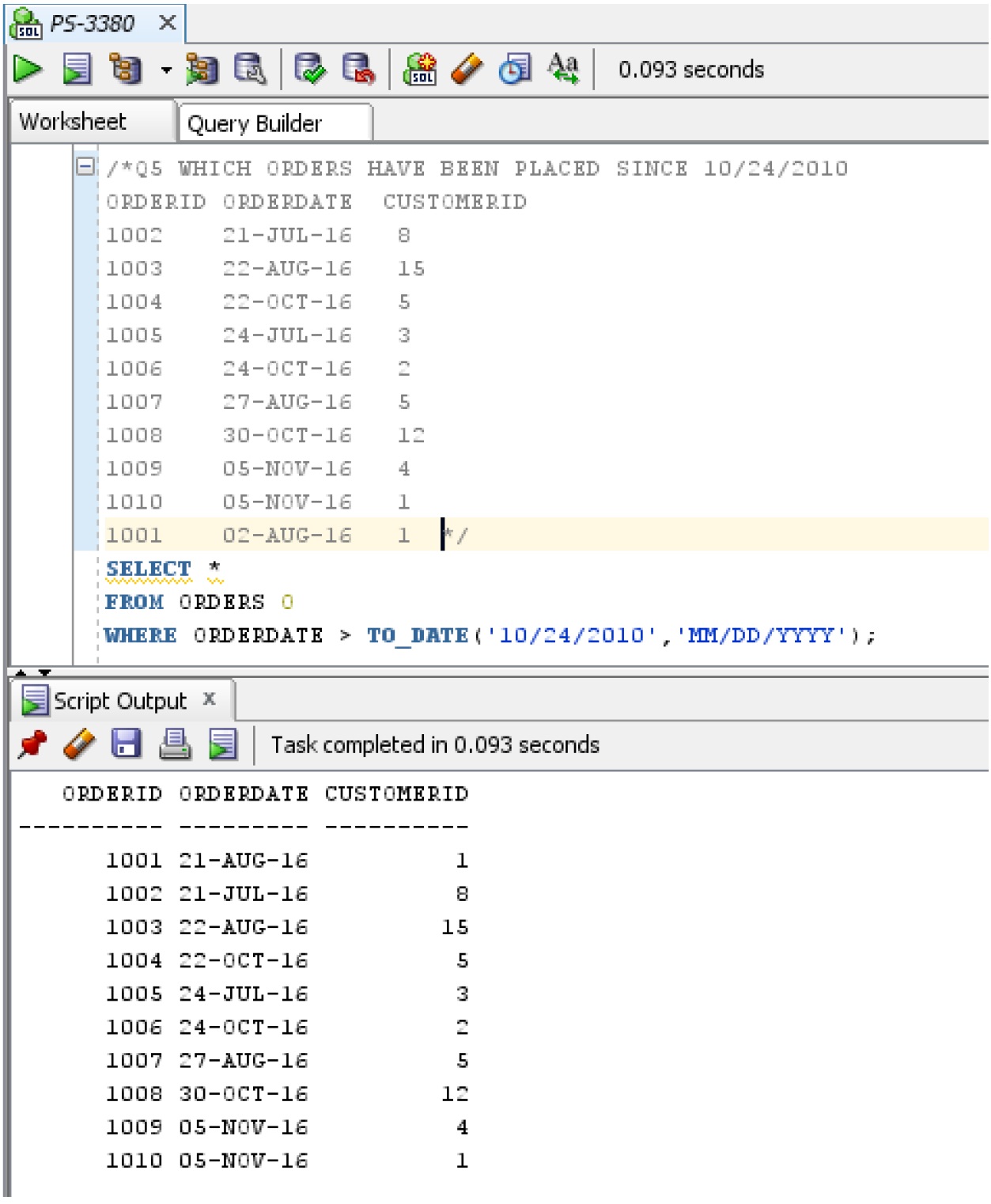
4. (3 points)

How many different items were ordered on order number 1004?



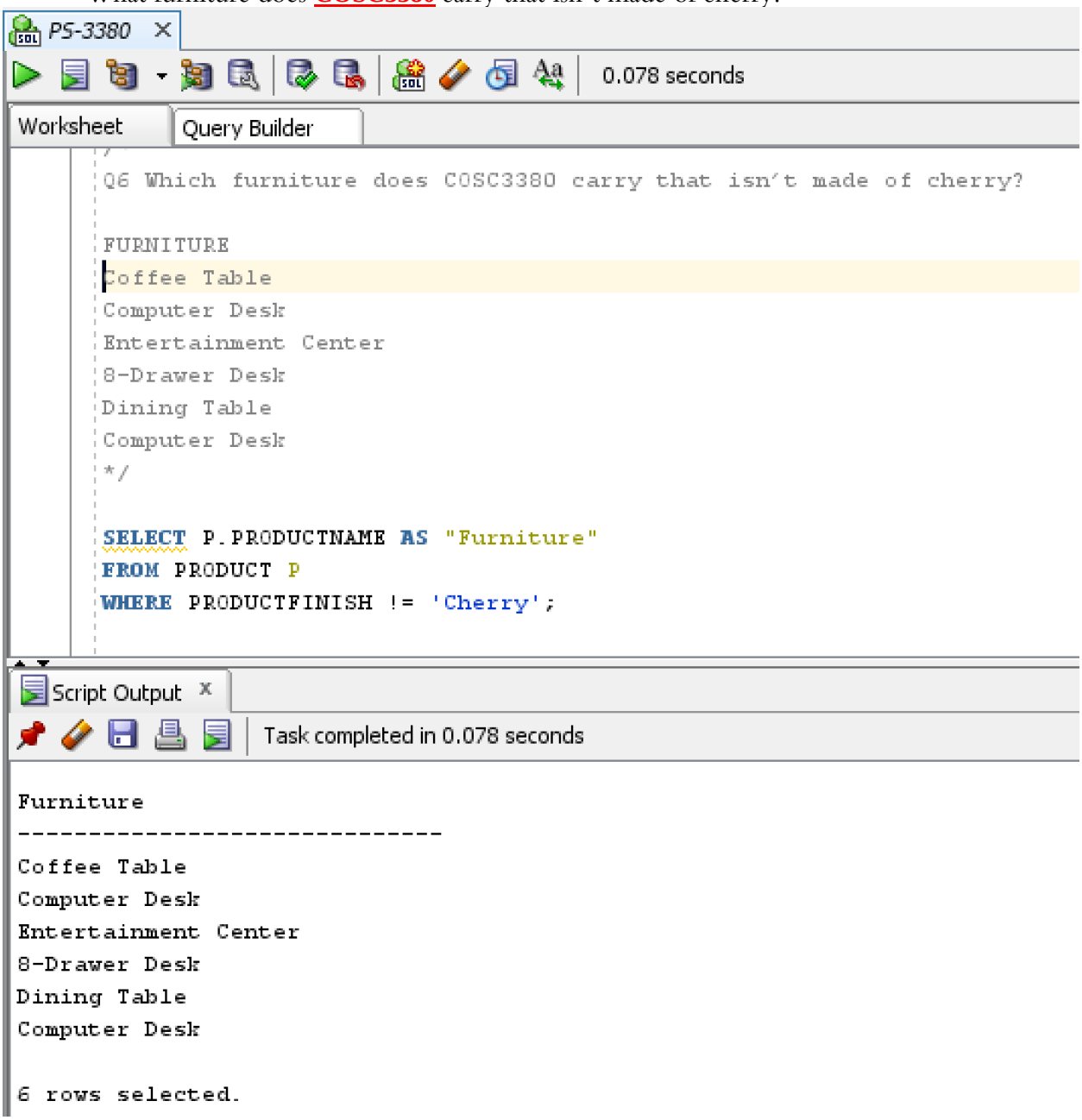
5. (3 points)

Which orders have been placed since 10/ 24/ 2010?


PS-3380
0.093 seconds
Worksheet
Query Builder
/*Q5 WHICH ORDERS HAVE BEEN PLACED SINCE 10/24/2010
ORDERID ORDERDATE CUSTOMERID
1002 21-JUL-16 8
1003 22-AUG-16 15
1004 22-OCT-16 5
1005 24-JUL-16 3
1006 24-OCT-16 2
1007 27-AUG-16 5
1008 30-OCT-16 12
1009 05-NOV-16 4
1010 05-NOV-16 1
1001 02-AUG-16 1 */
SELECT *
FROM ORDERS O
WHERE ORDERDATE > TO_DATE('10/24/2010','MM/DD/YYYY');
Script Output
Task completed in 0.093 seconds
ORDERID ORDERDATE CUSTOMERID
1001 21-AUG-16 1
1002 21-JUL-16 8
1003 22-AUG-16 15
1004 22-OCT-16 5
1005 24-JUL-16 3
1006 24-OCT-16 2
1007 27-AUG-16 5
1008 30-OCT-16 12
1009 05-NOV-16 4
1010 05-NOV-16 1

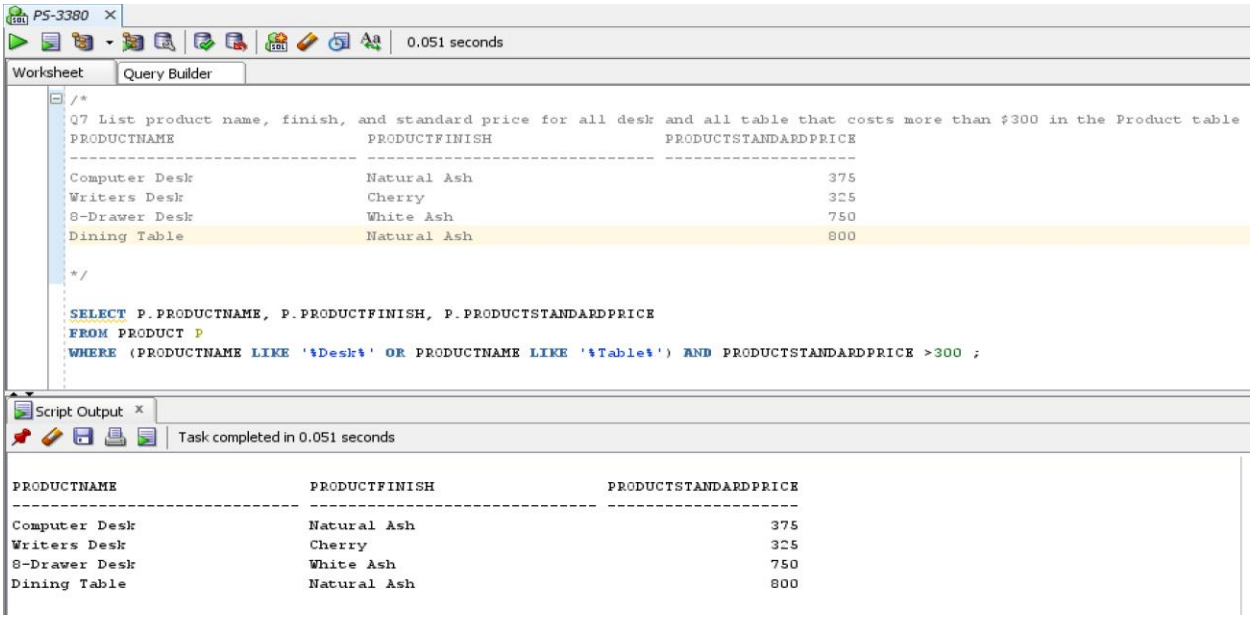
6. (3 points)

What furniture does **COSC3380** carry that isn't made of cherry?



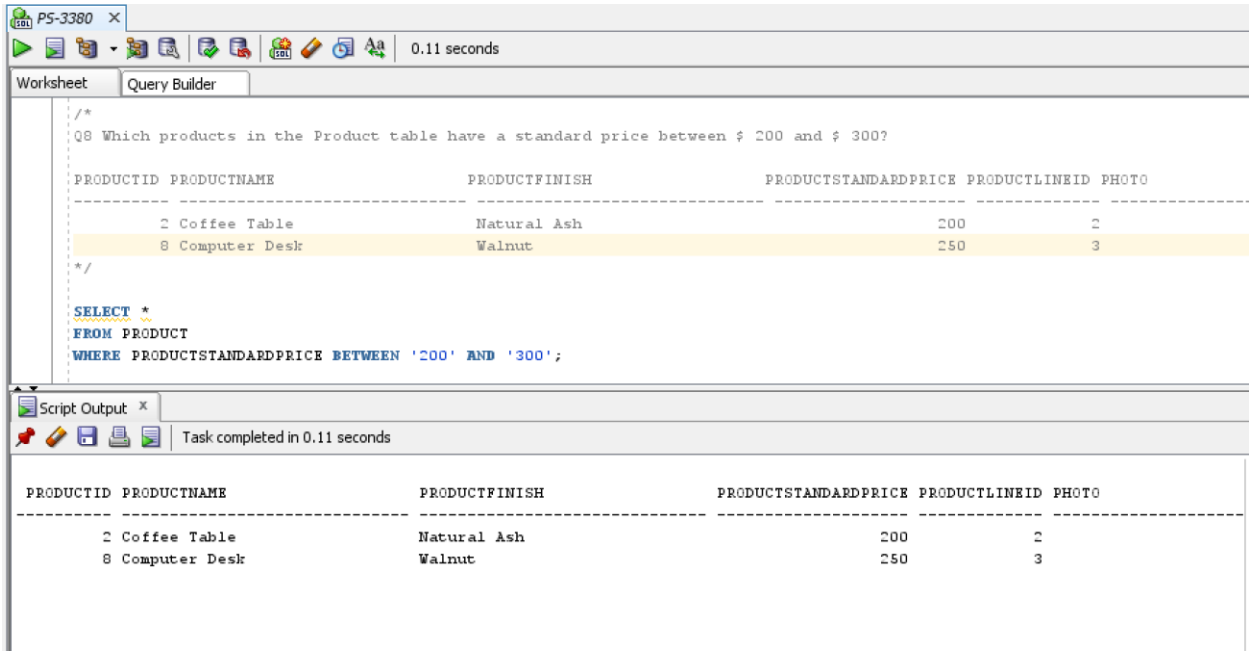
7. (3 points)

List product name, finish, and standard price for all desks and all tables that cost more than $ 300 in the Product table.



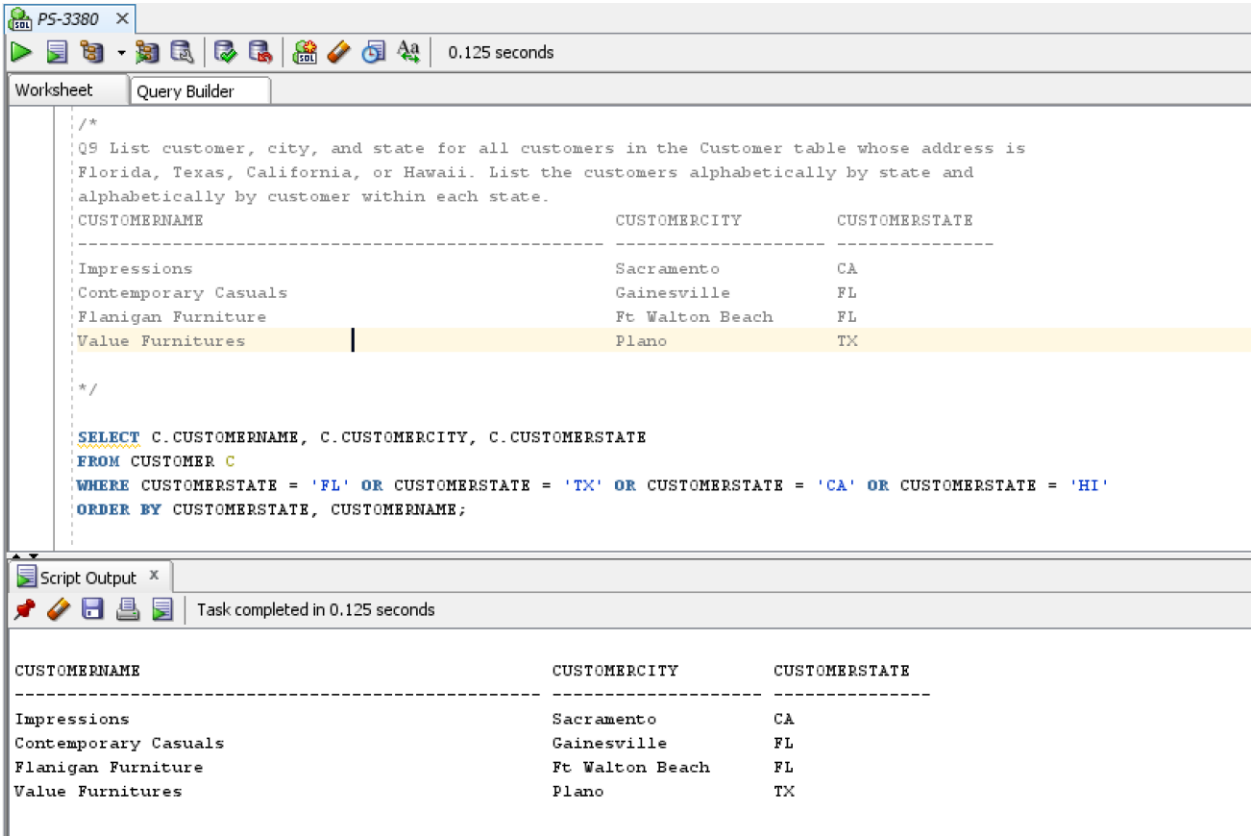
8. (3 points)

Which products in the Product table have a standard price between $ 200 and $ 300?



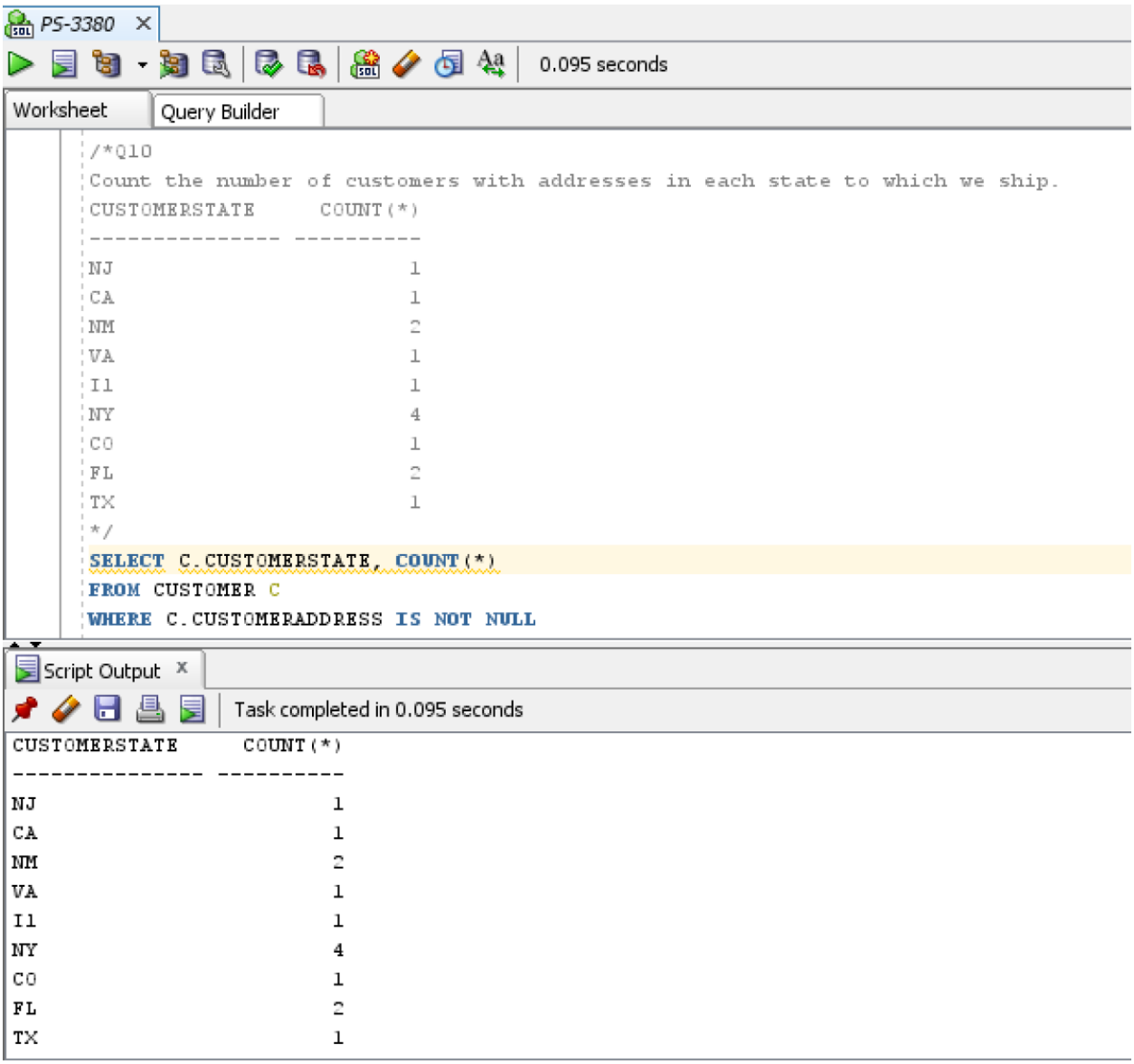
9. (3 points)

List customer, city, and state for all customers in the Customer table whose address is Florida, Texas, California, or Hawaii. List the customers alphabetically by state and alphabetically by customer within each state.



10. (3 points)

Count the number of customers with addresses in each state to which we ship.



11. (3 points)

Count the number of customers with addresses in each city to which we ship. List the cities by state.

PS-3380

Worksheet | Query Builder

```
/* Q11
Count the number of customers with addresses in each city to which we ship. List the cities by state.
CUSTOMERSTATE                                   CUSTOMERCITY                                    COUNT(*)
CA                                              Sacramento                                         1
CO                                              Boulder                                            1
FL                                              Ft Walton Beach                                    1
FL                                              Gainesville                                        1
Il                                              Oak Brook                                          1
NJ                                              Carteret                                           1
NM                                              Farmington                                         1
NM                                              Las Cruces                                         1
NY                                              Albany                                             1
NY                                              Rome                                               2
NY                                              Syracuse                                           1
TX                                              Plano                                              1
VA                                              Virginia Beach                                     1
*/
SELECT  DISTINCT C.CUSTOMERSTATE, C.CUSTOMERCITY, COUNT(*) AS "TOTAL COUNT"
FROM CUSTOMER C
GROUP BY CUSTOMERSTATE, CUSTOMERCITY
ORDER BY CUSTOMERSTATE, CUSTOMERCITY;
```

Query Result

SQL | All Rows Fetched: 13 in 0 seconds

	CUSTOMERCITY	CUSTOMERSTATE	TOTAL COUNT
1	Sacramento	CA	1
2	Boulder	CO	1
3	Ft Walton Beach	FL	1
4	Gainesville	FL	1
5	Oak Brook	Il	1
6	Carteret	NJ	1
7	Farmington	NM	1
8	Las Cruces	NM	1
9	Albany	NY	1
10	Rome	NY	2
11	Syracuse	NY	1
12	Plano	TX	1
13	Virginia Beach	VA	1

12. (3 points)

Find only states with more than one customer.

PS-3380

0.078 seconds

Worksheet | Query Builder

```
/* Q12
Find only states with more than one customer.

CUSTOMERSTATE          TOTAL
--------------- ----------
NM                          2
NY                          4
FL                          2
*/

SELECT C.CUSTOMERSTATE, COUNT(*) AS TOTAL
FROM CUSTOMER C
WHERE C.CUSTOMERADDRESS IS NOT NULL
GROUP BY CUSTOMERSTATE
HAVING COUNT(*) >1;
```

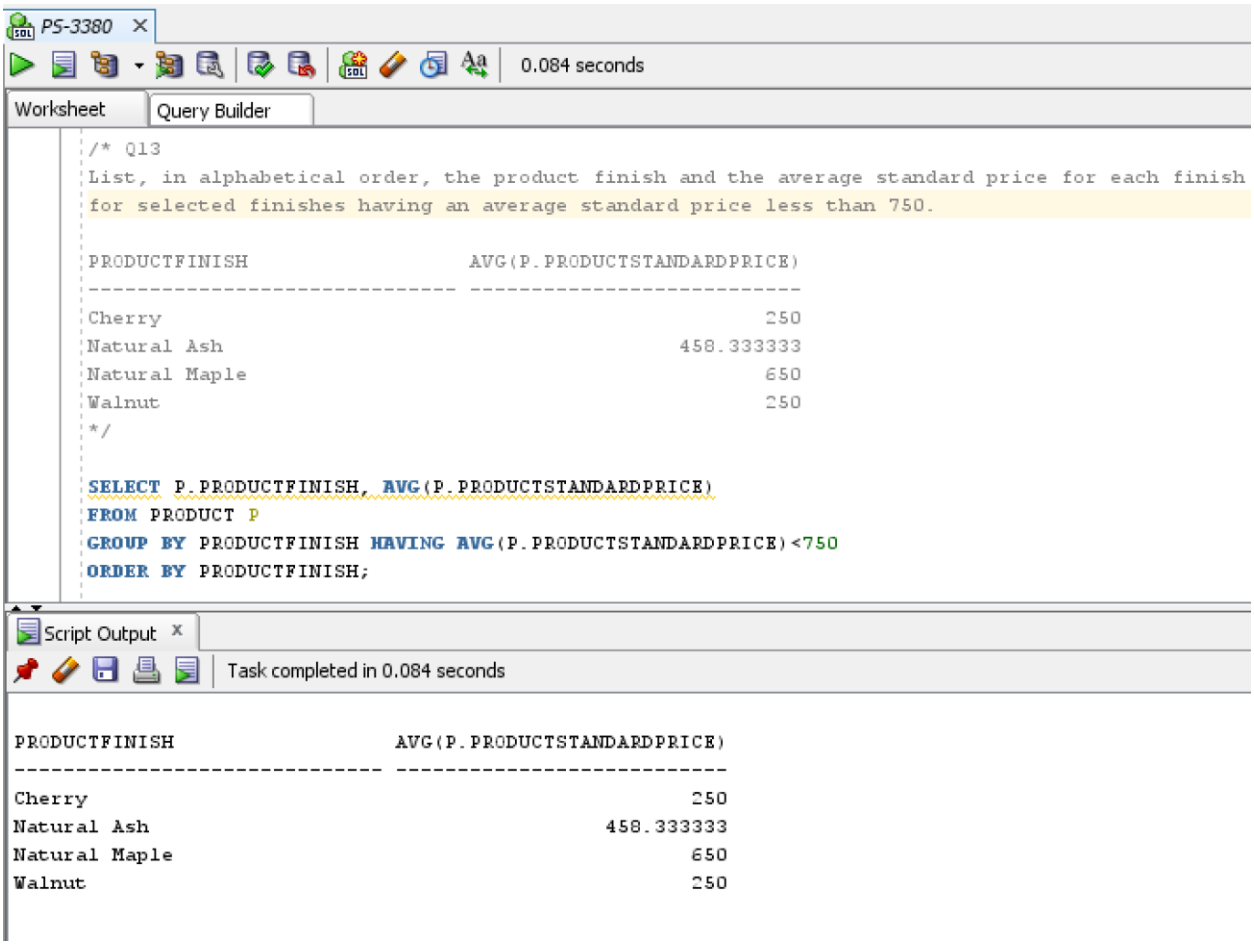
Script Output

Task completed in 0.078 seconds

```
CUSTOMERSTATE          TOTAL
--------------- ----------
NM                          2
NY                          4
FL                          2
```

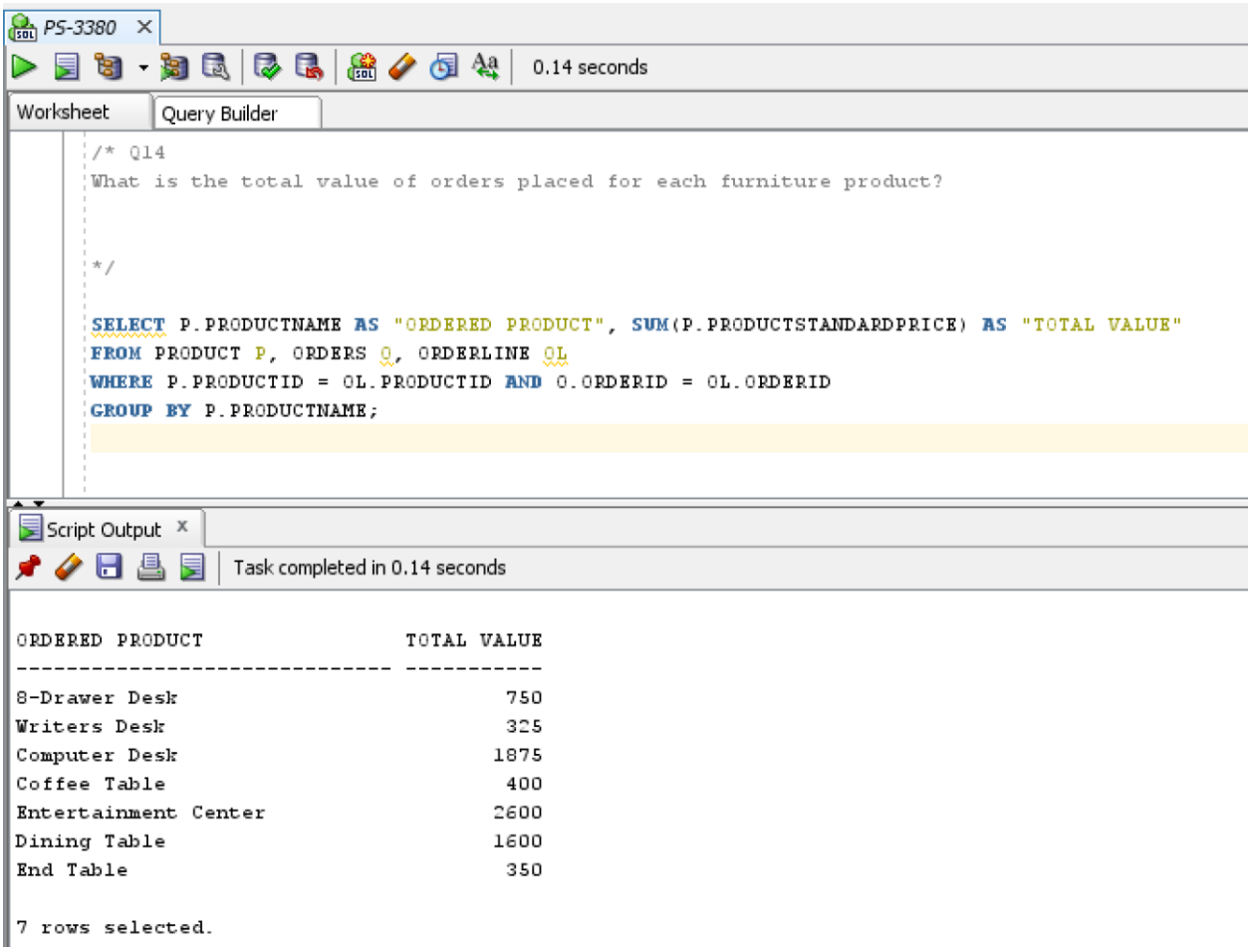
13. (3 points)

List, in alphabetical order, the product finish and the average standard price for each finish for selected finishes having an average standard price less than 750.



14. (3 points)

What is the total value of orders placed for each furniture product?


PS-3380
0.14 seconds
Worksheet
Query Builder
/* Q14
What is the total value of orders placed for each furniture product?
*/
SELECT P.PRODUCTNAME AS "ORDERED PRODUCT", SUM(P.PRODUCTSTANDARDPRICE) AS "TOTAL VALUE"
FROM PRODUCT P, ORDERS O, ORDERLINE OL
WHERE P.PRODUCTID = OL.PRODUCTID AND O.ORDERID = OL.ORDERID
GROUP BY P.PRODUCTNAME;
Script Output
Task completed in 0.14 seconds
ORDERED PRODUCT TOTAL VALUE
8-Drawer Desk 750
Writers Desk 325
Computer Desk 1875
Coffee Table 400
Entertainment Center 2600
Dining Table 1600
End Table 350
7 rows selected.

You need to turn in:

. The sql scripts with SQL COMMENTS for each query copied from this document and the expected output **Project 2.sql**. (Only to BB)

Example:

```
# Q1 Which products have a standard price of less than $ 275?
#       PRODUCTDESCRIPTION          PRODUCTSTANDARDPRICE
#       End Table                    175
#       Computer Desk                250
#       Coffee Table                 200
```

. The ***screenshots*** of SQL COMMENTS, SQL Oracle COMMANDS and the SQL OUTPUT in Project 2.doc. Please be paper efficient! (BOTH to BB and HARDCOPY)

Without Project 2.doc you will get ZERO!

Tips for the Projects:

1) Prepare your SQL program using the Text editor such as Note Pad **Project 2.sql**.
2) Copy and Paste your program from **Project 2.sql** to SQL*Plus command line.
3) If the program works, save the program. Otherwise, fix the error for the program in **Project 2.sql** and try it again.
4) Capture screenshots (CROP if needed) of the SQL COMMENTS, SQL COMMAND, and SQL OUTPUT and place them in Project 2.doc.
5) Submit **Project 2.sql** that contains all your SQL COMMENTED SQL commands to BB.
6) Submit **Project 2.doc** that contains all your screenshots to BB and HARDCOPY.
7) Fill in the **Cover Sheet** and you are ready.